IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 17, 2018, accepted April 20, 2018, date of publication May 1, 2018, date of current version June 5, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2832077

Secure Delegation-Based Authentication for Telecare Medicine Information Systems

ZUOWEN TAN

School of Information Technology, Jiangxi University of Finance and Economics, Nanchang 330032, China
Nanchang Institute of Science and Technology, Nanchang 330108, China
Key Laboratory of Information Security, School of Mathematics and Information Science, Guangzhou University, Guangzhou 510006, China

e-mail: tanzyw@163.com

This work was supported in part by the National Natural Science Foundation of China under Grant 61462033, in part by the Natural Science Foundation of Jiangxi Province under Grant 20161BAB202059, in part by the Science and Technology Project of the Provincial Education Department of Jiangxi under Grant GJJ160430, and in part by the Open Project Program of the Guangdong Provincial Key Laboratory of Information Security under Grant GDXXAQ2016-10.

ABSTRACT The telecare medicine information systems (TMISs) enable patients to gain health monitoring at home and obtain medical services over mobile networks. In recent years, many authentication schemes have been proposed to address the security and privacy issues in the TMISs. For example, Kim *et al.* and Huang *et al.* proposed efficient delegation-based authentication protocols by using elliptic curve cryptography. These protocols have a prerequisite that both the home location register and the visited location register must share secrets beforehand. In this paper, we show that Kim *et al.*'s and Hwang *et al.*'s schemes are vulnerable to known key attacks. Moreover, they fail to provide communication confidentiality. We then present a new secure delegation-based authentication protocol by using the identity-based cryptography. The proposed protocol removes the weaknesses of the above-mentioned protocols. Through the analysis of the Burrows–Abadi–Needham logic, along with a random oracle model, we demonstrate that the proposed scheme provides secure authentication. In addition, the proposed scheme can provide more security functionaries than the existing delegation-based authentication protocols. Better tradeoff among security and functionality features and communication and computation costs makes our scheme suitable and applicable in the TMISs.

INDEX TERMS Authentication, BAN, cryptography, delegation.

I. INTRODUCTION

Telecare is playing a role in the medical service. The telecare medicine information systems (TMISs) have many advantages over the traditional healthcare systems. Through the universal roaming technology, legitimate mobile users enjoy wireless services. Even if the users are far from the medical local center, they can still gain access to the healthcare services from the medical center over mobile networks. However, since the communication channel in TMISs is open, the security issues have always been a major concern. In order to obtain data integrity and confidentiality, authentication and key agreement schemes are essential for TMISs [1]–[3]. The authentication mechanism of TMISs involves three entities, a mobile user (MU), a visited location register (VLR) and a home location register (HLR). MU first registers at HLR. When MU roams into a foreign network and tries to login a VLR, the VLR validates the user's legality with the help of the HLR. Cryptographic techniques are always used to establish authentication mechanisms. There are two kinds of cryptographic systems: private key cryptosystem and public key cryptosystem. The encryption/decryption operation of the private key cryptosystem is faster than that of public key cryptosystem, but the private key cryptosystem cannot provide nonrepudiation. The most widely used authentication protocol based on the private key cryptosystem is GSM [1] which adopts the encryption/decryption algorithm A5. GSM has low computational loads for MU. Moreover, it only needs simple key management. However, the secret-key cryptosystem based authentication system cannot provide the nonrepudiation or privacy protection of mobile users. Lee and Yeh [4] pointed that MU cannot authenticate VLR in GSM.

According to the literature [5]–[8], a authentication protocol for wireless communications should hold the following security and functionality requirements.

SECURITY REQUIREMENTS

(S1) Resistance to DoS attacks. Even if an adversary has mounted login or authentication request many times within

a short time, the authentication between the legitimate MU and the VLR is still available.

(S2) Resistance to request replication attacks. For the authentication protocols, the replication attack leads to the accounting problem. The attempts to intercept the messages between communicating parties and replay these messages in the further processes should be prevented.

(S3) Resistance to impersonation attacks. An adversary always fails in impersonating as a legitimate user to fool the HLR/VLR, or impersonating as the HLR/VLR to communicate with the legitimate user.

(S4) Resistance to known-key attacks. A protocol is called secure against known-key attacks if the adversary which has obtained several shared keys of past sessions cannot still compromise a shared key of current session. A secure wireless communication system ensures that the succeeding encrypted message cannot be extracted via the present session keys.

FUNCTIONALITY REQUIREMENTS

(F1) Mutual authentication. It means that each entity should authenticate each other. VLR and HLR ensure that MU is indeed a registered user of HLR. MU believes that the registration confirmation message is indeed from the HLR. Both VLR and MU believe that HLR is indeed the registration server of MU. Both HLR and MU authenticate VLR which MU attempts to visit.

(F2) Nonrepudiation. Once the dispute happens during the online authentication phase, HLR or MU cannot deny the access request of MU. Once the dispute occurs during the offline authentication phase, MU cannot deny the fact that he/she has gained access to VLR.

(F3) Privacy protection. User's privacy is always an imperative issue in the TMISs. It is always maintained through user anonymity support. In cellular networks, GSM and 3GPP roaming protocols provide a certain degree of anonymity by using some temporary identity called TMSI (Temporary Mobile Subscriber Identity) rather than the real identity IMSI (International Mobile Subscriber Identity). Preserving user anonymity and un-traceability is fundamental for protection of user privacy protection in wireless communications.

According to the extent of the privacy protection, we classify the user privacy protection in TMISs into the following types: anonymity, weak un-traceability, and strong un-traceability.

Anonymity: No one except the HLR can reveal the user's real identity during roaming. The user's identity cannot be obtained by the eavesdroppers, other mobile users, and even the foreign servers.

Weak un-traceability: Any unauthorized entities cannot track the mobile user's movements. The concept ''weak'' means that the HLR or VLR may link two different sessions to a same MU. Different sessions of the same user within one foreign domain can be easily linked by the VLR. However, any outside adversary cannot trace the MU from the message transmitted over open channels.

Strong un-traceability: It is the concatenation of user un-traceability and anonymity. The real identity of MU should not be comprised during online authentication phase or offline authentication phase. Even the VLR cannot link two different sessions to a same MU or a same identify after MU was involved in any previous protocol runs.

(F4) Communication confidentiality. After MU and VLR have established a session key, their communication message cannot be compromised by any other entities including the HLR, which is called communication confidentiality. It is necessary to ensure the privacy of the MU and the VLR in roaming services.

A. OUR CONTRIBUTIONS

In this paper, we propose a novel delegation-based authenticated key agreement protocol (hereafter we call it as a DBAKA protocol) for wireless roaming service based on identity-based cryptography [9], which satisfies more security requirements and functionality requirements than the existing DBAKA protocols for wireless roaming service.

In summary, the following contributions are listed below:

- A concept ''communication confidentiality'' is introduced. We argue that communication confidentiality is necessary after the mobile user and the VLR have finished establishing a session key with help of the HLR. The feature has never been considered in the existing authentication schemes for roaming network.
- We analyze two DBAKA schemes for wireless roaming service [8], [10] and reveal their weaknesses. We found that these protocols [8], [10] suffer from known session key attacks.
- Analysis of formal security under Random Oracle model and Burrows-Abadi-Needham (BAN) logic shows that the proposed scheme provides secure authentication.
- The proposed scheme provides better security as compared with the other relevant DBAKA schemes for wireless roaming service. And it removes the weaknesses of the protocols [8], [10], [11].
- Our DBAKA protocol for wireless roaming service does not require pre-shared keys between the VLR and HLR. It can be used to offer secure and expeditious services with the reasonable computational, communication, and storage overhead.

B. MOTIVATION

In a DBAKA protocol for roaming network, the property of being resisting against known-key attacks must be considered carefully. There are always the scenarios: when MU has realized that one session key has been revealed, maybe MU has already accessed VLR several times. Especially for offline authentication phase, it is indeed vital that the information cannot be acquired from the previous transmitted message by using some present session keys (*not limited a single session key*). The security model should cover the situation: even if an adversary has obtained all the participants' secret keys, it still cannot recover other session keys from a set of known consecutive or discontinuous session keys. Till date, none of

DBAKA protocols for wireless roaming service delivers a mechanism to resist known session key attacks from consecutive or discontinuous session keys.

To the best of my knowledge, the fairness of the session key among the MU, the VLR and the HLR is overemphasized. The communication confidentiality between the MU and the VLR has never been studied in the existing DBAKA protocols for wireless roaming service. In fact, if the session key is acquired by the HLR, neither of the MU's privacy and the VLR's privacy cannot be well protected.

This motivates us to design a DBAKA protocol for wireless roaming service which both resists against known session key attacks and holds the communication confidentiality.

C. SYSTEM MODEL

1) NETWORK MODEL

The proposed DBAKA protocol involves a mobile user MU, a visited location register VLR and a home location register HLR. At the setup phase, a trusty authority (Private Key Generator, PKG) derives private keys from arbitrary public keys for the HLRs and HLRs. Before getting access to wireless service, each mobile user registers at the HLR. The HLR assigns a proxy key to each registered user. All the participants have information about the public parameters of all the HLRs and VLRs, including their identities.

2) ADVERSARY MODEL

As in [4], we assume that HLR is a trusty entity. Only the HLR can reveal the real identity of MU. In our model, HLR is honesty but curious, which has not been discussed in the literature. Since the HLR may want to know about the service with which VLR provides MU, the assumption is reasonable. In addition, when MU registers with the HLR, the channel between MU and HLR is secure. In the system, a passive adversary eavesdrops on all the authentic messages among the participants and may relay them. An active adversary even can modify messages and insert messages over the channel among the MU, the HLR and the VLR.

Our DBAKA protocol provides mobile users with weak un-traceability. The real identity of the mobile user is not used to attain mutual authentication between the VLR and MU. Any adversary cannot trace the MU from the message transmitted over open channels.

D. ORGANIZATION OF THE PAPER

The remainder of this paper is organized as follows. Section II briefly reviews the existing authentication schemes for roaming networks. In Section III, we give cryptanalysis of two DBAKA protocols for wireless roaming service [8], [10]. In Section IV, we propose a new DBAKA protocol. Security and functionality analysis and performance comparison of our scheme with related existing schemes is given in Section V and Section VI. Finally, we conclude the paper in Section VII.

II. RELATED WORK

Over last few years, researchers have developed public-key system based authentication protocols to remove the weaknesses of the authentication protocols based on private key cryptosystem and provide the MU with the privacy protection [12]–[14]. However, since MUs have to retrieve the most recent certificate revocation list and update public keys periodically, MUs require high computation cost. Due to the hardware limitations of the portable devices [15], MU cannot support too complicated encryption or decryption operations. Another problem of the public key cryptosystem based authentication protocols is that certificate will compromise real identity of MU.

Recent years have witnessed the efforts on the privacy protection [16], [17]. Various anonymous authentication methods are proposed [18], [19]. According to the number of participants, these authentication protocols could be divided into two types: three-party protocols involving a home server, two-party protocols without a home server. Yang *et al.* [20], He *et al.* [21] and Jo *et al.* [22] proposed anonymous two-party roaming protocols, respectively. However, Zhao *et al.* [23] pointed out that Mun *et al.*'s scheme [24] cannot withstand replay attacks, man-in-the-middle attacks, and insider attacks. Since these protocols are based on group signature or signcryption, they require high computational cost at the mobile devices. Moreover, the home server and foreign server are assumed to share a roaming key. The protocol [24] requires that the HLR is online during the authentication process. In such a situation, HLR always becomes a bottleneck [25], [26].

In order to increase efficiency, the concept of delegation is applied to design light weight authentication protocols [4], [8]. The idea of delegation is inspired by the proxy signature which is generated by the proxy signer after the original signer has delegated his signature authority to the proxy signer [27]. When one verifies a proxy signature, the public key of the original signer will be used. Thus, the original signer cannot deny the proxy signature. In 2005, Lee and Yeh [4] proposed an anonymous delegation based authentication protocol. In Lee and Yeh's protocol [4], HLR and MU act as the original signer and the proxy signer, respectively. Since VLR verifies the signature via the public key of HLR, VLR does not know the real identity of MU. Thus, the protocol in [4] achieves the user anonymity. In addition, it achieves nonrepudiation and mutual authentication between MU and VLR. The protocol uses offline authentication process to reduce the communication overhead among the VLR, HLR and MU. In the DBAKA protocol for wireless roaming service, the authentication process is divided into two phases, online authentication and offline authentication. During the online authentication phase, VLR has authenticated MU with help of the HLR, while VLR fulfills authentication without contacting HLR during the second phase. Moreover, the offline authentication process has as high communicational

efficiency as in GSM [1]. The DBAKA protocol has the merit of both three-party authentication protocols and two-party authentication protocols. However, Tang and Wu [7] pointed out that Lee and Yeh's DBAKA protocol suffers from the VLR impersonation attack. Lee *et al.* [28] showed that Lee and Yeh's offline authentication process is vulnerable to masquerade user attacks. Any legal VLR can forge login message of a mobile user during the offline authentication phase. Moreover, the Lee and Yeh's DBAKA protocol cannot provide user non-repudiation during the offline authentication phase [28]. Two enhanced DBAKA protocols for wireless roaming service use backward hash chains to remove the weaknesses [7], [28]. Unfortunately, Youn and Lim [29] demonstrated that Lee *et al.*s protocol [28] cannot achieve weak untraceability. Furthermore, Lee *et al.*'s protocol cannot provide forward secrecy [30], [31]. Lee *et al.* [30] found that neither of Lee and Yeh's [4] and Lee *et al.*'s [28] protocols achieves weak anonymity or forward secrecy. Lu and Zhou [32], [33] pointed out that Tang and Wu's protocol [7] is vulnerable to the replication attack. It still cannot provide weak untraceability [32]. Wang and Lin [34] showed that Youn and Lim's DBAKA protocol for wireless roaming service [29] suffers from Denial of Service (DoS) attack. Wang *et al.* [35] demonstrated that Youn and Lim's DBAKA protocol for wireless roaming service [29] cannot provide the weak untraceability. Recently, Gope and Hwang [31] showed that the improved protocol [30] suffers from certain weaknesses like the vulnerability to DoS attack, no perfect forward secrecy, loss of untraceability, etc. Recently, Tsai *et al.* [11] used ECC to present a DBAKA protocol that requires less computation cost. Like the other DBAKA protocols [4], [7], [28]–[30], [32]–[35], Tsai et al.'s protocol requires that the VLR and HLR must pre-share some secrets. Kim *et al.* [8] demonstrated that Tsai et al.'s protocol suffers from known session key attacks. They presented an improved version. Hwang and You [10] adopt the embedded concurrent signcryption scheme to design another delegation-based authentication protocol.

III. CRYPTANALYSIS OF DBAKA PROTOCOLS

In this section, we first tabulate the important notations in Table 1. We then review two DBAKA protocols for wireless roaming service and show that they fail to provide known key security and communication confidentiality.

A. BRIEF REVIEW OF KIM et al.'s PROTOCOL

Kim et al. disclosed the vulnerability of Tsai et al.'s delegation-based authentication protocol [11] and then presented a solution to eliminate the potential threat [8]. The improvement on offline authentication is described as follows.

- **Step 1.** During the i-th offline authentication phase, MS issues $E_{C_i}[h^{(n-i+1)}(n_1), E_{y_v}(h^{(n-i-1)}(n_1))]$ to VLR, where $E_{y_v}()$ is asymmetric encryption with VLR's public key.

TABLE 1. The notations.

Symbol	Description
q	large prime number
G_1	a cyclic additive group of order q
G_2	a multiplicative group of order q
P	a generator of Group G_1
$\hat{e}$	a bilinear map: $G_1 \times G_1 \rightarrow G_2$
$E_k(M)$	a symmetric encryption of message M with k
ID_V/ID_H	the identity of VLR/the identity of HLR
$h()$	a one-way hash function: $\{\}^* \rightarrow Z_q$

- **Step 2.** VLR decrypts the message from MS and obtains $h^{(n-i+1)}(n_1)$, $E_{y_v}(h^{(n-i-1)}(n_1))$.
- **Step 3.** VLR decrypts $E_{y_v}(h^{(n-i-1)}(n_1))$ with the private key xv and then verifies whether $h(h(h^{(n-i-1)}(n_1))) = h^{(n-i+1)}(n_1)$. If the above equality holds, VLR calculates the session key $C_{i+1} = h(h^{(n-i-1)}(n_1), C_i)$.

B. SECURITY WEAKNESSES OF KIM et al.'s PROTOCOL

Kim *et al.* [8] argue that these modifications described in Subsection A of Section III can effectively prevent the known key attacks because any adversary can compute neither the token $h^{(n-i)}(n_1)$ nor $h^{(n-i-1)}(n_1)$ from $h^{(n-i+1)}(n_1)$ due to the hash function's one-way characteristics. However, we demonstrate that the weaknesses of Tsai et al.'s protocol still exist in Kim et al.'s protocol.

1) KNOWN SESSION KEY ATTACKS

Assume that an adversary has obtained two session keys $\{C_i, C_{i+2}\}$ during the offline authentication phase. Further assume the adversary has intercepted MU's login message $E_{C_i}[h^{(n-i+1)}(n_1), E_{y_v}(h^{(n-i-1)}(n_1))]$ transmitted during the $(i+1)$-th offline authentication phase. An adversary can mount the known session key attack as follows.

First the adversary uses the key C_i to decrypt the ciphertext and obtains $h^{(n-i+1)}(n_1)$. With knowledge of session key C_{i+2}, the adversary computes session key $C_{i+3} = h(h^{(n-i+1)}(n_1), C_{i+2})$, and then computes session key $C_{i+4} = h(h(h^{(n-i+1)}(n_1)), C_{i+3})$, etc.

2) LACK OF COMMUNICATION CONFIDENTIALITY

During online authentication phase, HLR has obtained $(N_1 \| n_2 \| K)$ as described in Subsection A of Section III. Thus HLR can compute the session key SK. Likewise, HLR can apply the technique as described in the known key attack to compute all the session keys for offline authentication. Even if the HLR is not a malicious entity, the HLR may still be curious about the service with which the VLR provides the MU. Hence, with knowledge of the session key, the HLR

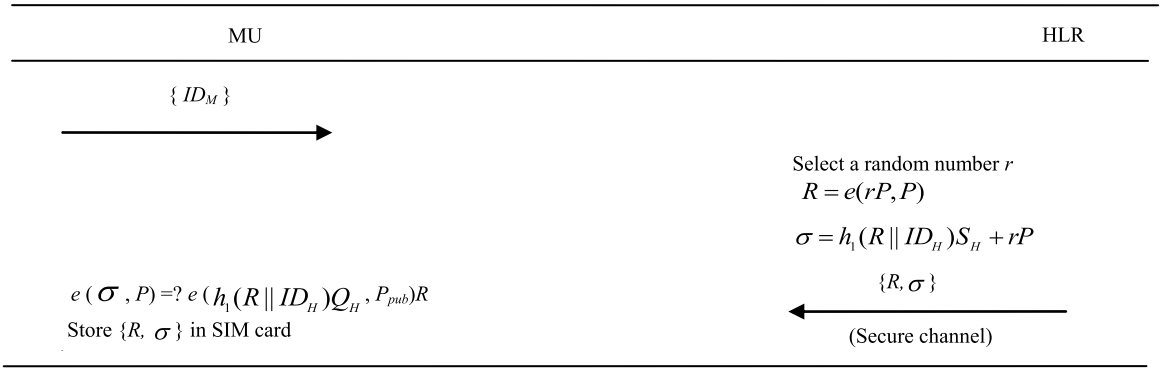


FIGURE 1. Registration phase.

obtains the communication message between the mobile user and the VLR.

C. BRIEF REVIEW OF HWANG et al.'s PROTOCOL

During the offline authentication phase of Hwang et al.'s DBAKA protocol for wireless roaming service [10], VLR authenticates MU repeatedly without contacting HLR. Let RK_i be the session key of i-th subsequent login.

- **Step 1.** MU transmits $(ID_i, E_{RK_i}(h^{(n-i+1)}(n_1)))$ to VLR.
- **Step 2.** VLR searches its local database through ID_i and decrypts $E_{RK_i}(h^{(n-i+1)}(n_1))$.
- **Step 3.** VLR sets $L = h^{(n-i+1)}(n_1)$ and calculates

$$RK_{i+1} = h(L, RK_i), \quad ID_{i+1} = H(ID_i, RK_{i+1}).$$

- **Step 4.**VLR computes $E_{RK_i}(ACK)$ and transmits it to User, where $ACK = h(h^{(n-i+1)}(n_1), ID_i)$.
- **Step 5.** MU decrypts $E_{RK_i}(ACK)$ and validates ACK.

D. WEAKNESSES OF HWANG et al.'s DBAKA PROTOCOL

In the following, we show that the Hwang et al.'s protocol suffers from known session key attacks. Therefore it lacks communication confidentiality. Thus, once one session key is compromised, MU can be tracked.

1) KNOWN SESSION KEY ATTACKS

Assume that an adversary has obtained one session key RK_i. Further assume the adversary has intercepted login message $(ID_i, E_{RK_i}(h^{(n-i+1)}(n_1)))$ transmitted during the $(i+1)$-th offline authentication phase. With knowledge of RK_i, the adversary decrypts the ciphertext and obtains $h^{(n-i+1)}(n_1)$. Thus, the adversary can compute a new session key $RK_{i+1} = h(L, RK_i)$ where $L = h^{(n-i+1)}(n_1)$.

2) LACK OF COMMUNICATION CONFIDENTIALITY

Through the similar analysis to that of Kim et al.'s DBAKA protocol for wireless roaming service, we can show that Hwang et al.'s DBAKA protocol cannot provide communication confidentiality.

3) LACK OF USERS' UNLINKABILITY

If one session key, say RK_i, is compromised, according to the above analysis, the new session key RK_{i+1} can be obtained. Thus, by checking $ID_{i+1} = H(ID_i, RK_{i+1})$, one can decide whether two logins are from a same user where one login message contains ID_i while the other login message contains ID_{i+1}.

IV. THE PROPOSED DBAKA PROTOCOL

In this section, we propose a novel DBAKA protocol for wireless roaming service based on identity-based cryptography. The proposed protocol is composed of five phases: setup, registration, online authentication, offline authentication and revocation phase.

A. SETUP PHASE

PKG performs the setup process which is presented below.

- **Step 1.** Generate an additive group G_1 of prime order q (e.g., 160 bits) with a generator P, a multiplicative group G_2 of prime order q and a bilinear map e:$G_1 \times G_1 \rightarrow G_2$, and choose a random $s \in Z_q^*$ as the master secret key and compute $P_{pub} = sP$ as the public key.
- **Step 2.** Select cryptographic hash functions: H: $\{0,1\}^* \rightarrow G_1$, h_1: $\{0,1\}^* \rightarrow Z_q^*$, h_2: $\{0,1\}^* \rightarrow \{0,1\}^{l_1}$, h_3: $\{0,1\}^* \rightarrow \{0,1\}^{l_2}$, h_4: $\{0,1\}^* \rightarrow \{0,1\}^{l_3}$, h_5: $\{0,1\}^* \rightarrow \{0,1\}^{l_4}$, h_6: $\{0,1\}^* \rightarrow \{0,1\}^k$, where k is the security parameter.
- **Step 3.** Compute $Q_V = H(ID_V)$, $Q_H = H(ID_H)$, $S_V = sQ_V$ and $S_H = sQ_H$. Then PKG issues the key (Q_V, S_V), (Q_H, S_H) for the VLR and HLR, respectively. VLR and HLR validate their private keys by verifying the equations, respectively:

$$e(S_V, P) = e(Q_V, P_{pub}), \quad e(S_H, P) = e(Q_H, P_{pub}). \tag{1}$$

The system public parameters are $\{G_1, G_2, q, P, e, H(), h_i(), i = 1, 2, \ldots, 6, P_{pub}\}$.

B. REGISTRATION PHASE

MU performs the following steps to register with the HLR as described in Fig. 1.

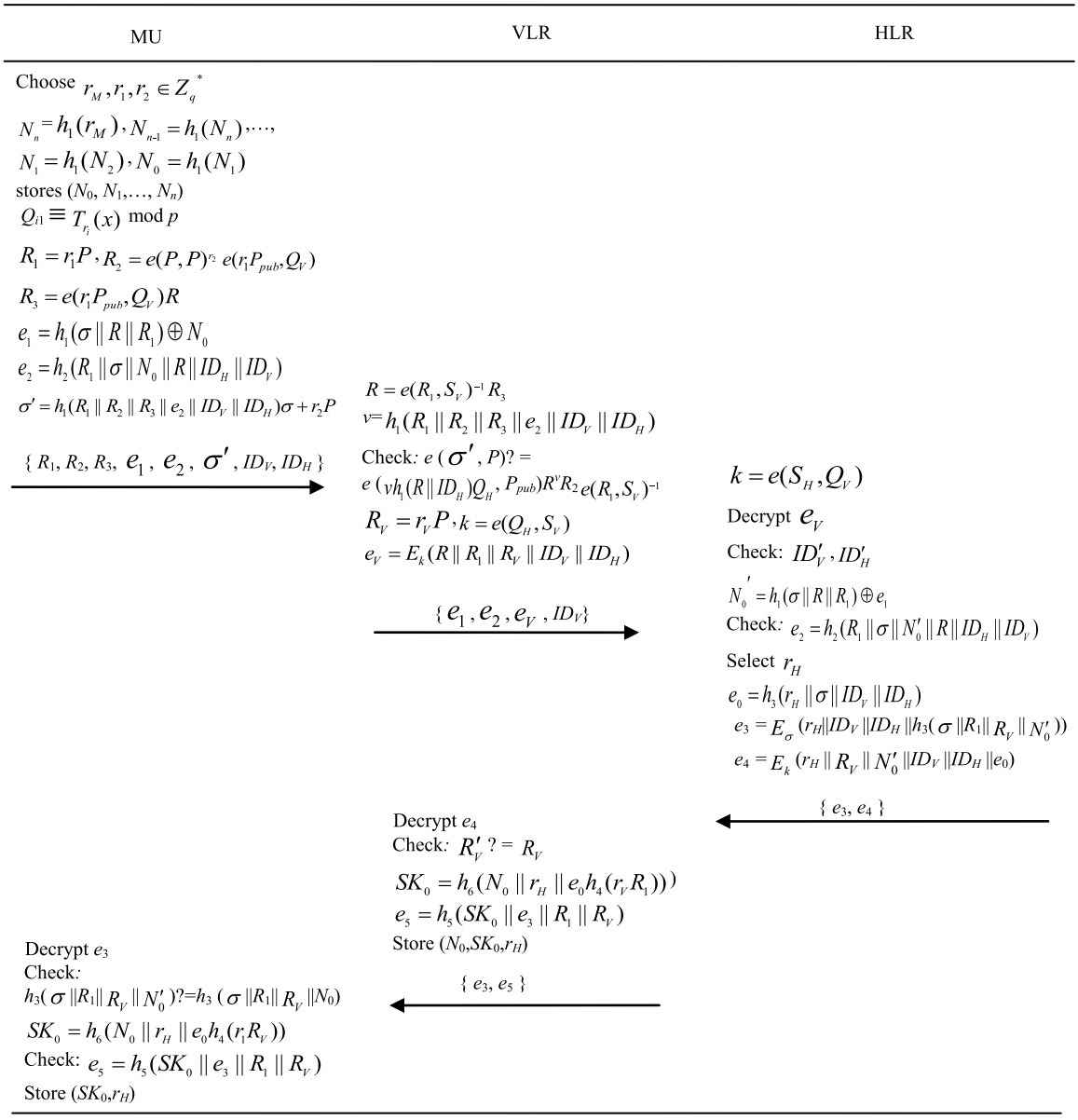


FIGURE 2. Online authentication and key agreement phase.

- **Step 1.** MU issues $\{ID_M\}$ to HLR.
- **Step 2.** HLR randomly chooses a number $r \in Z_q^*$ and computes a proxy key (R, σ)

$$R = e(rP, P), \quad \sigma = h_1(R \| ID_H) S_H + rP. \tag{2}$$

- **Step 3.** HLR sends (R, σ) to MU through a secure channel. HLR stores (ID_M, R, σ) in a list where ID_M is the identity of MU.
- **Step 4.** MU checks its validity through the equation:

$$e(\sigma, P) = e(h_1(R \| ID_H) Q_H + rP, P_{pub}) R. \tag{3}$$

Afterwards, the key pair (R, σ) is stored in MU's SIM card.

C. ONLINE AUTHENTICATION PHASE

MU and VLR cooperatively establish an authenticated key with the participation of HLR. We describe the online authentication phase which is also shown in Fig. 2.

- **Step 1.** When MU tries to launch an online authentication, MU sends a login request to VLR. MU selects a positive integer n as the total number of offline authentication, chooses a random number $r_M \in Z_q^*$ and computes the hash values

$$N_n = h_1(r_M), \quad N_{n-1} = h_1(N_n), \ldots, N_1 = h_1(N_2),$$
$$N_0 = h_1(N_1).$$

MU stores the hash chain $(N_0, N_1, \ldots, N_n)$ in the SIM card.

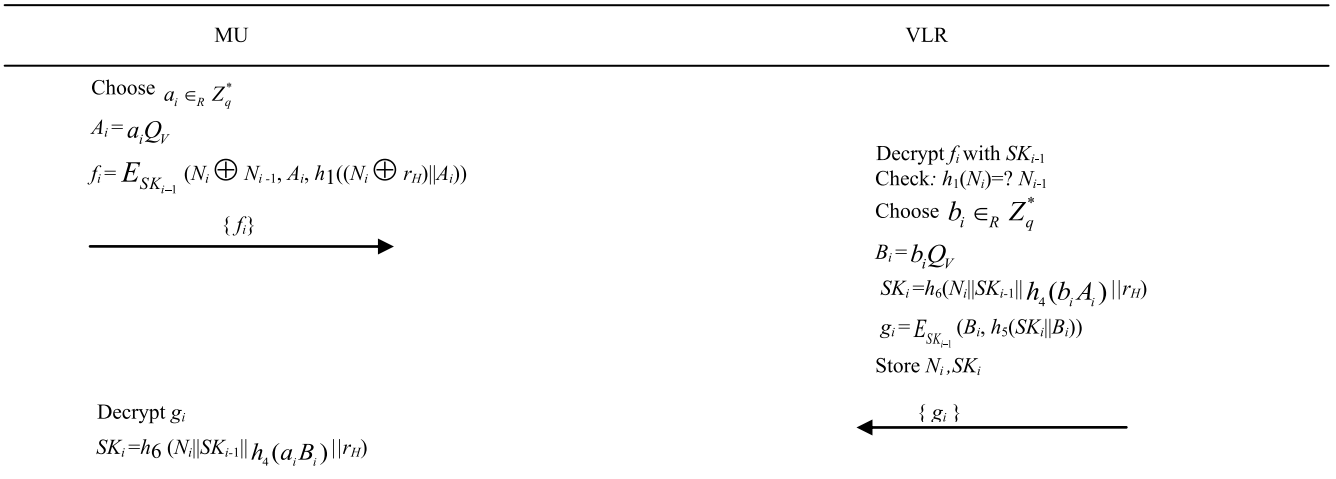


FIGURE 3. Offline authentication and key agreement phase.

- **Step 2.** MU chooses two random numbers $r_1, r_2 \in Z_q^*$ and calculates

$$R_1 = r_1 P, \quad R_2 = e(P, P)^{r_2} e(r_1 P_{pub}, Q_V), \tag{4}$$
$$R_3 = e(r_1 P_{pub}, Q_V) R, \quad e_1 = h_1(\sigma \| R \| R_1) \oplus N_0, \tag{5}$$
$$e_2 = h_2(R_1 \| \sigma \| N_0 \| R \| ID_H \| ID_V), \tag{6}$$
$$\sigma' = h_1(R_1 \| R_2 \| R_3 \| e_2 \| ID_V \| ID_H)\sigma + r_2 P. \tag{7}$$

  Then MU sends $\{R_1, R_2, R_3, e_1, e_2, \sigma', ID_V, ID_H\}$ to VLR.
- **Step 3.** VLR recovers

$$R = e(R_1, S_V)^{-1} R_3, \; v = h_1(R_1 \| R_2 \| R_3 \| e_2 \| ID_V \| ID_H)$$

  and verifies if the equation holds:

$$e(\sigma', P) = e(v h_1(R \| ID_H) Q_H, P_{pub}) R^v R_2 e(R_1, S_V)^{-1}. \tag{8}$$

  If the equation holds, VLR randomly generates a number $r_V \in Z_q^*$, and computes

$$R_V = r_V P, \quad e_V = E_k(R \| R_1 \| R_V \| ID_V \| ID_H), \tag{9}$$

  where $k = e(Q_H, S_V)$. Finally, VLR sends $\{e_1, e_2, e_V, ID_V\}$ to HLR.
- **Step 4.** HLR computes $k = e(S_H, Q_V)$ and decrypts e_V to obtain $\{R, R_1, R_V, ID'_V, ID'_H\}$. Then HLR checks if ID'_V and ID'_H are the same as the received identity and its identity, respectively. If they both are valid, HLR searches the database for (*, R, *) and obtains σ. HLR computes $N'_0 = h_1(\sigma \| R \| R_1) \oplus e_1$, and verifies if the equation $e_2 = h_2(R_1 \| \sigma \| N'_0 \| R \| ID_H \| ID_V)$ holds. If it holds, HLR generates a random number r_H, and computes

$$e_0 = h_3(r_H \| \sigma \| ID_V \| ID_H), \tag{10}$$
$$e_3 = E_\sigma(r_H \| ID_V \| ID_H \| R_V \| h_3(\sigma \| R_1 \| R_V \| N'_0)), \tag{11}$$
$$e_4 = E_k(r_H \| R_V \| N'_0 \| ID_V \| ID_H \| e_0). \tag{12}$$

  Finally, HLR appends (ID_V, r_H) behind (ID_M, R, σ) in its database and sends $\{e_3, e_4\}$ to VLR.
- **Step 5.**VLR decrypts e_4 and obtains $\{r_H, R'_V, N'_0, ID_V, ID_H, e_0\}$. VLR checks if the received R'_V is the same as R_V, and if ID_V and ID_H are valid. After they have been verified, VLR computes

$$SK_0 = h_6(N_0 \| r_H \| e_0 \| h_4(r_V R_1)), \tag{13}$$
$$e_5 = h_5(SK_0 \| e_3 \| R_1 \| R_V). \tag{14}$$

  Then VLR stores (N_0, SK_0, r_H) and sends $\{e_3, e_5\}$ to MU. Otherwise, VLR refuses the login request from MU.
- **Step 6.** MU uses σ to decrypt e_3 and obtain $\{r_H, ID_V, ID_H, R_V, h_3(\sigma \| R_1 \| R_V \| N'_0)\}$. MU checks whether the decrypted strings ID_H and ID_V are equal to HLR's identity and VLR's identity, respectively. If either is invalid, MU refuses the response from VLR. Otherwise, MU checks if $h_3(\sigma \| R_1 \| R_V \| N'_0) = h_3(\sigma \| R_1 \| R_V \| N_0)$. If it holds, MU computes $SK_0 = h_6(N_0 \| r_H \| e_0 h_4(r_1 R_V))$ and checks whether $e_5 = h_5(SK_0 \| e_3 \| R_1 \| R_V)$. If it holds, MU stores (SK_0, r_H) in the SIM card.

D. OFFLINE AUTHENTICATION PHASE

MU and VLR cooperatively perform offline authentication which does not require HLR online. We describe the offline authentication phase as follows, which is also shown in Fig. 3.

- **Step 1.** When MU wants to issue the i-th ($i = 1, 2, \ldots, n$) offline authentication request, MU first picks (N_i, SK_{i-1}) in the SIM card. Then MU chooses a random number $a_i \in_R Z_q^*$ and calculates $A_i = a_i Q_V$, $f_i = E_{SK_{i-1}}(N_i \oplus N_{i-1}, A_i, h_1((N_i \oplus r_H)\|A_i))$ and sends f_i to VLR.
- **Step 2.** VLR decrypts f_i with last session key SK_{i-1} and obtains N_i by using the stored N_{i-1}. Then VLR checks whether $h_1(N_i)$ is the same as the stored value N_{i-1} and the third plaintext is $h_1((N_i \oplus r_H)\|A_i)$. If they are equal,

VLR chooses a random number $b_i \in_R Z_q^*$ and calculates

$$B_i = b_i Q_V, \quad SK_i = h_6(N_i \| SK_{i-1} \| h_4(b_i A_i) \| r_H), \tag{15}$$

$$g_i = E_{SK_{i-1}}(B_i, \quad h_5(SK_i \| B_i)). \tag{16}$$

VLR replaces N_i with N_{i-1}, stores the key SK_i and removes SK_{i-1}. Finally VLR issues g_i to MU.

- **Step 3.** Upon receiving the response from VLR, MU decrypts g_i and obtains B_i. Then MU computes $SK_i = h_6(N_i \| SK_{i-1} \| h_4(a_i B_i) \| r_H)$. MU checks if the second part of plaintext is equal to $h_5(SK_i \| B_i)$. If so, MU stores the session key SK_i.

The offline authentications have been performed until i is equal to n. Then MU launches a new online authentication with VLR and HLR by another request.

E. REVOCATION PHASE

The HLR will suspend MU's service or revoke the MU when either of the following three cases happens: (1) The key of the mobile user is comprised; (2) The mobile device is stolen/lost; (3) MU' service subscription expires. If MU's service is suspended, all VLRs will not provide MU with the roaming service. In order to attain the objective, HLR periodically sends VLR the list of suspended users which consist of $E_k(r_H)$. VLR decrypts it and checks if r_H is in the service list. Thus, when a revoked user tries to establish a session key and sends f_i to VLR, the VLR can identify the user by checking the validity of $h_1((N_i \oplus r_H) \| A_i)$.

When HLR revokes a certain MU, HLR removes the MU's account by simply putting the key (R, σ) in the revocation list without keeping his/her identity unchanged. Simultaneously, the HLR informs the VLR about this by sending $E_k(r_H)$ with the revocation message.

V. SECURITY ANALYSES

In this section, through the formal and informal security analysis, we show that our scheme can resist various known attacks.

A. FORMAL SECURITY ANALYSIS IN RANDOM ORACLE MODEL

Definition 1: Let G_1 be an elliptic curve group over finite field F_p and P is a point of large prime order q in G_1. Given (P, aP, bP) in G where unknown $a, b \in Z_q^*$, how to compute abQ is called **Elliptic curve Diffie–Hellman problem (ECDH problem)**.

The success probability of a probabilistic polynomial time Turing machine Δ in solving ECDH problems in G_1 is defined as:

$$\mathrm{Succ}_G^{CDH}(\Delta) = \Pr[\Delta\,(aQ, bQ) = abQ : a, b \in_R Z_q^*].$$

Definition 2 (The Elliptic Curve Diffie–Hellman (ECDH Assumption): is the assumption that it is infeasible to solve ECDH problems within polynomial time. In other words, for every probabilistic polynomial time Turing machine Δ, $\mathrm{Succ}_G^{CDH}(\Delta)$ is negligible.

Since Abdalla *et al.* [36] (hereafter called as AFP security model) is appropriate for three-party authenticated key agreement scenario, we will give the security proof of the proposed scheme in AFP security model. By U, V, and S, we denote mobile user, visited location register and home location register, respectively. The ith instance of U is denoted by U^i. The jth instance of V is denoted by V^j and the kth instance of S is denoted by S^k. An adversary A is modeled as a probabilistic polynomial time Turing machine. A can issue a bounded number of the following queries:

Send ($U^i/V^j/S^k$, Q): The adversary A sends message Q to instance $U^i/V^j/S^k$, and the instance $U^i/V^j/S^k$ gives a reply according to the protocol. The *Send* query modles active attack.

Reveal (U^i/V^j): If no session key is defined for instance U^i/V^j, or if either U^i/V^j or its partner has been asked a *Test* query, the oracle outputs the invalid symbol $\perp$. Otherwise the oracle outputs the current session key SK generated by U^i/V^j (and its partner) to A. The *Reveal* query models known session key attack.

Corrupt (U/V): This query returns to the adversary A the proxy key pair for participant U or the private key for participant V.

Test (U^i/V^j): If no session key is defined for instance U^i/V^j or if either U^i/V^j or its partner has been issued a *Reveal* query, the oracle will output the invalid symbol $\perp$. Otherwise, the oracle flips a coin b. If $b = 1$, the oracle outputs the session key. Otherwise, the oracle returns a random string drawn from the space of session keys. Note that *Test* query of this form is allowed to be invoked to a fresh oracle once by the adversary.

Note that the query *Execute* is not described. In essence, in the AFP security model, the query *Execute* can be simulated by using the *Send* queries repeatedly on condition that there exists at least one benign adversary.

The AFP security model defines the semantic security by a game of two phases. During the first phase, A is allowed to adaptively issue *Send*, *Reveal* and *Test* queries. During the second phase, A executes a single *Test* query with chosen bit b to a fresh instance and the query outputs a guess bit b' for b. If $b' = b$, then the adversary A wins the game.

Definition 3: Let $Succ(A)$ be the event that the adversary A wins the above game, i.e., A is successful in breaking the semantic security of the DBAKA protocol for wireless roaming service. The advantage of the adversary A in breaking the **semantic security** of our protocol by guessing the correct bit b' is defined by

$$Adv_G^{DBAKA}(A) = |2\Pr(Succ(A)) - 1| = |2\Pr[b' = b] - 1|.$$

A DBAKA protocol is said to be semantically secure in AFP security model if the advantage of any probabilistic polynomial time-bounded adversary A is negligible.

Theorem 1: Let A be a polynomial time bounded adversary. Assume that hash functions $h_i()(i=1,2,3,4,5,6)$ are modeled

TABLE 2. Simulation of hash, reveal, test, and corrupt oracle queries.

Hash simulation query performs as follows:
On a hash query H_i (i=1,2,…,6} oracle about m
if a record (i,m, h) exists in list L_i , **then**
Return hash value h.
else Select a string $h \leftarrow_R Z_q^*$ for i=1; $h \leftarrow_R \{0,1\}^{l_i}$, for i=2,..5; $h \leftarrow_R \{0,1\}^k$, for i=6;
if the query is initiated by the adversary A, **then**
The triple (i,m, h) is added to L_A.
else Add (i,m, h) into L_i.
end if
end if

Reveal(U^i/V^j) simulation query performs as follows:
if session key SK is defined for instance U^i or V^j, **then**
The output of this query is SK.
else The output is $\perp$.
end if

Corrupt(U/V) simulation query performs as follows: On a *Corrupt*(U) query, return the proxy key pair (R, σ) of participant U as the output of the query; On a *Corrupt*(V) query, return the private key S_V of participant V as the output of the query.

Test(U^i/V^j) simulation query performs as follows:
By using *Reveal*(U^i/V^j) query, obtain the output of *Reveal* query.
if the output is $\perp$, **then**
Return $\perp$ as the output of *Test*(U^i/V^j) query.
else The oracle flips a unbiased coin b.
if b = 1,**then**
Return the session key SK as the output of the *Test* query
else Return a random string from $\{0,1\}^k$ as the output of the *Test* query.
end if
end if

as random oracles. Assume that A is allowed to make at most q_s times Send queries and at most q_i times hash oracle H_i (i=1,2,3,4,5,6) query, respectively. Then, we have

$$Adv_G^{DBAKA}(A) \le \frac{3q_s^2}{2^{k+1}} + \frac{q_s^2+q_1^2}{2(q-1)} + \sum_{i=2}^{6}\frac{q_i^2}{2^{i_i+1}} + \frac{q_s}{q-1} + \frac{3q_2+2q_s}{2^{i_2}} + \frac{2q_3}{2^{i_3}} + \frac{2q_4+q_s}{2^{i_4}} + \frac{q_5}{2^{i_5}} + q_7\mathrm{Succ}_G^{CDH}(A),$$

where $Adv_G^{DBAKA}(A)$ is the advantage of the adversary A in breaking the semantic security of our protocol and $\mathrm{Succ}_G^{CDH}(A)$ is the success probability of a probabilistic polynomial time Turing machine in solving ECDH problems.

Proof: We define sequent games to prove this theorem, $\boldsymbol{G_i}(i = 0, 1, 2, 3, 4)$. Let $Succ_i$ be an event defined as successful guessing of the bit b in *Test* query corresponding to each game $\boldsymbol{G_i}$ by an adversary A.

Game $\boldsymbol{G_0}$: This starting game is the actual attack game as in the real protocol. Hence, we have

$$Adv_G^{DBAKA}(A) = |2Pr[Succ_0] - 1|. \tag{17}$$

Game $\boldsymbol{G_1}$: This game simulates all oracle queries including *Send*, *Reveal*, *Corrupt*, *Test* and *hash* queries. We give the simulation of the hash oracles and *Reveal*, *Corrupt*, *Test* queries in Table 2. We simulate the *Send* queries (see Table 3) as in the actual attack game. The simulations maintain lists for queries: (1) list L_i answers hash oracle $H_i(i = 1, 2, \ldots, 6)$, (2) list L_A records queries which are initiated by A.

Since this game is perfectly indistinguishable from the actual attack game, we have

$$\Pr[Succ_1] = \Pr[Succ_0]. \tag{18}$$

Game $\mathbf{G_2}$: In this game, we consider the probability of collisions among the results of the hash oracle queries that the adversary asks H_1 query, H_2 query, and random numbers in the transcripts of messages transmitted in our protocol. We take the random value h from Z_q^* for H_1 and $\{0, 1\}^k$ for H_2. If this query is directly asked by the adversary, and $(i,^*, h) \in L_i(i = 1, 2, 3, 4, 5, 6)$, we abort the game. Otherwise, h is returned. Since that hash value h is chosen uniformly at random, according to the birthday paradox, the probability of collisions is at most $\frac{q_1^2}{2(q-1)}$. Further, messages transmitted in our protocol contain randomly chosen elements $\{R_1, R_2, R_3, R_V, r_H\}$, and the probability of randomly chosen elements collision is at most $\frac{q_s^2}{2^{k+1}} + \frac{2q_s^2}{q-1}$.

Games $\mathbf{G_2}$ and $\mathbf{G_1}$ are perfectly indistinguishable unless the above-mentioned collisions cause the simulator to abort the game. Hence, we have

$$|\Pr(Succ_2) - \Pr(Succ_1)| \le \frac{q_s^2}{2^{k+1}} + \frac{4q_s^2+q_1^2}{2(q-1)}. \tag{19}$$

Game $\boldsymbol{G_3}$: This game considers a situation where A obtains the correct message transcript luckily without the query about

TABLE 3. Simulation of send oracle queries.

The following hash values in Table III are generated as per Table II. From now on, we will omit the statement in the context.

– On a query *Send* (U^i, Start), we proceed as follows:

Compute $R_1 = r_1 P$, $R_2 = e(P,P)^{r_2} e(r_1 P_{pub}, Q_V)$, $R_3 = e(r_1 P_{pub}, Q_V) R$, $e_1 = h_1(\sigma \| R \| R_1) \oplus N_0$,

$e_2 = h_2(R_1 \| \sigma \| N_0 \| R \| ID_H \| ID_V)$, $\sigma' = h_1(R_1 \| R_2 \| R_3 \| e_2 \| ID_V \| ID_H) + r_2 P$.

Return $\{R_1, R_2, R_3, e_1, e_2, \sigma', ID_V, ID_H\}$. Then, instance U^i proceeds to an expecting state.

– On a query *Send* (U^i, e_3, e_5), we proceed as follows if instance U^i is in an expecting state:

Decrypt e_3 with σ and check whether $h_3(\sigma \| R_1 \| R_V \| N_0') = h_3(\sigma \| R_1 \| R_V \| N_0)$.

if the equation does not hold, **then**

instance U^i terminates without accepting

else instance U^i accepts. Compute $SK = h_6(N_0 \| r_H \| e_0 h_4(r_1 R_V))$ and check whether $e_5 = h_5(SK_0 \| e_3 \| R_1 \| R_V)$.

if the equation holds, **then**

instance U^i terminates.

end if

end if

– On a query *Send* (V^j, R_1, R_2, R_3, e_1, e_2, σ', ID_V, ID_H), we proceed as follows if instance V^j is in an expecting state:

Compute $R = e(R_1, S_V)^{-1} R_3$, $v = h_1(R_1 \| R_2 \| R_3 \| e_2 \| ID_V \| ID_H)$ and check whether $e(\sigma', P) = e(v h_1(R \| ID_H) Q_H, P_{pub}) R R_2 e(R_1, S_V)^{-1}$.

if the equation holds, V^j accepts, **then**

We proceed as follows. Compute $R_V = r_V P, e_V = E_k(R \| R_1 \| R_V \| ID_V \| ID_H)$, Return $\{e_1, e_2, e_V, ID_V\}$. Then the instance V^j goes onto an expecting state.

else Reject the response

end if

– On a query *Send* (V^j, e_3, e_4), we proceed as follows if instance V^j is in an expecting state:

Decrypt e_4 and check whether R_V' is equal to R_V.

if they are not equal, **then**

Instance V^j terminates without accepting.

else Instance V^j accepts. We proceed as follows: Calculate $SK = h_6(N_0 \| r_H \| e_0 h_4(r_V R_1)), e_5 = h_5(SK_0 \| e_3 \| R_1 \| R_V)$.

Return $\{e_3, e_5\}$. Instance V^j terminates.

end if

– On a query *Send* (S^k, e_1, e_2, e_V, ID_V), we proceed as follows:

Compute $k = e(S_H, Q_V)$ and decrypt e_V. Compute $N_0' = h_1(\sigma \| R \| R_1) \oplus e_1$ and check whether $e_2 = h_2(R_1 \| \sigma \| N_0' \| R \| ID_H \| ID_V)$.

if the equation does not hold, **then**

Instance S^k terminates.

else Instance S^k accepts. And we proceed as follows: compute $e_0 = h_3(r_H \| \sigma \| ID_V \| ID_H)$, $e_3 = E_\sigma(r_H \| ID_V \| ID_H \| h_3(\sigma \| R_1 \| R_V \| N_0'))$,

$e_4 = E_k(r_H \| R_V \| N_0' \| ID_V \| ID_H \| e_0)$. Return $\{e_3, e_4\}$. Then the instance S^k goes onto an expecting state.

end if

– On a query *Send* (U^i, offline,Start), we proceed as follows:

Select $a_i \in_R Z_q^*$, and compute $A_i = a_i Q$, $_i = E_{SK_{i-1}}(N_i \oplus N_{i-1}, A_i, h_1((N_i \oplus r_H) \| A_i))$. Return $\{f_i\}$.

Then the instance U^i goes onto an expecting state.

– On a query *Send* (U^i, g_i), we proceed as follows if instance U^i is in an expecting state:

Decrypt g_i and compute $SK_i = h_6(N_i \| SK_{i-1} \| h_4(a_i B_i) \| r_H)$. Instance U^i terminates.

– On a query *Send* (V^j, f_i), we proceed as follows if instance V^j is in an expecting state:

Decrypt f_i with SK_{i-1}, and check whether $h_1(N_i) = N_{i-1}$.

if the equation does not hold, **then**

Instance V^j terminates.

else We proceed as follows: choose $b_i \in_R Z_q^*$, calculate $B_i = b_i Q$, $SK_i = h_6(N_i \| SK_{i-1} \| h_4(b_i A_i) \| r_H)$, $g_i = E_{SK_{i-1}}(B_i, h_5(SK_i \| B_i))$.

Return $\{g_i\}$. Then the instance V^j goes onto an expecting state.

end if

hash oracles H_1. Online and offline authentication phases in our protocol contain six messages m_i, ($i = 1, 2, ..6$) where by the symbols m_1, m_2, m_3, m_4, m_5 and m_6, we denote the messages $\{R_1, R_2, R_3, e_1, e_2, \sigma', ID_V, ID_H\}$, $\{e_1, e_2, e_V, ID_V\}$, $\{e_3, e_4\}$, $\{e_3, e_5\}$, $\{f_i\}$ and $\{g_i\}$, respectively. We discuss it in the following queries, *Send*(U, m_1), *Send*(V,m_2), *Send*(S,m_3), *Send*(V,m_4), *Send*(U, m_5), and *Send*(V, m_6) query. Take *Send*(U, m_1) query for example. We consider the maximum probability of the hash values falling within the list L_A, which is involved in the message m_1. As per the

protocol, the values $h_1(\sigma\|R\|R_1)$ and $h_1(R_1\|R_2\|R_3\|e_2\|ID_V\|ID_H)$ must be in L_A, otherwise the session will be terminated. The maximum calculated probability is up to $\frac{2q_1}{q-1}$.

Furthermore, $(R_1\|\sigma\|N_0\|R\|ID_H\|ID_V), e_2)$ must be in L_A, whose probability is at most $\frac{q_2}{2^{i_1}}$. Hence, the probability in this case is at most $\frac{2q_1}{q-1}+\frac{q_2}{2^{i_1}}$. Considering all the cases, we have

$$|\Pr(Succ3)-\Pr(Succ_2)| \leq \frac{3(q_1+q_s)}{q-1}+\frac{q_2}{2^{i_1}}+\frac{q_3}{2^{i_2}}+\frac{q_4}{2^{i_3}}+\frac{q_5+q_s}{2^{i_4}}. \quad (20)$$

Game G_4: In this game, we replace random oracle H_6 with private oracle H_7. Assume that we do not use the $h_4(r_1R_V)$ or $h_4(r_VR_1)$ to compute session key SK. Thus, the session key is completely independent of H_7 and either $h_4(r_1R_V)$ or $h_4(r_VR_1)$. Thus, the session key is determined without querying the hash oracle. Since H_7 is a private oracle, the probability that adversary A correctly guesses the value of b in the game is

$$\Pr(Succ_4)=1/2. \quad (21)$$

Games $\mathbf{G_3}$ and $\mathbf{G_4}$ are perfectly indistinguishable unless the following event AskH occurs: the adversary A queries hash function H_6 on $N_0\|r_H\|e_0\|h_4(r_VR_1)$ or on $N_0\|r_H\|e_0\|h_4(r_1R_V)$. Hence, we have

$$|\Pr(Succ_4)-\Pr(Succ_3)| \leq \Pr(\mathrm{AskH}_6). \quad (22)$$

Now, we estimate the probability Pr (AskH_6). According to the definition of event AskH_6, the event AskH_6 means that the adversary has issued random oracle H_6 queries on $(N_0, r_H, e_0, \mathrm{CDH}(R_1, R_V))$. Since the number of records in the list L_6 is q_6, the probability of obtaining the $\mathrm{CDH}(R_1, R_V)$ value from list L_6 is $1/q_6$. Hence, we get

$$\Pr(\mathrm{AskH}_6)=q_6\mathrm{Succ}_G^{CDH}(A). \quad (23)$$

Using the triangular inequality, (18), and (21), we have the following:

$$\begin{aligned}&|\Pr[Succ_0]-1/2|\\ &=|\Pr[Succ_1]-\Pr[Succ_4]|\\ &\leq|\Pr[Succ_1]-\Pr[Succ_2]|+|\Pr[Succ_2]-Pr[Succ_3]|\\ &\quad+|\Pr[Succ_3]-\Pr[Succ_4]|. \quad (24)\end{aligned}$$

From Equations (17)-(24), we get

$$\begin{aligned}&Adv_G^{DBAKA}(A)\\ &\leq\frac{q_s^2}{2^{k+1}}+\frac{4q_s^2+q_1^2}{2(q-1)}+\frac{3(q_1+q_s)}{q-1}\\ &\quad+\frac{q_2}{2^{i_1}}+\frac{q_3}{2^{i_2}}+\frac{q_4}{2^{i_3}}+\frac{q_5+q_s}{2^{i_4}}+q_6\mathrm{Succ}_G^{CDH}(A). \quad (25)\end{aligned}$$

Thus, we have completed the proof of the theorem.

B. AUTHENTICATION PROOF BASED ON BAN-LOGIC

We will apply the well-popular BAN-logic [37] to validate the proposed protocol. The formal verification using BAN logic demonstrates that the proposed protocol achieves mutual authentication and it allows a user to establish a session key with the server. It is well known that BAN logic is a widely used logical formal model analysis method of reasoning the beliefs of participants in an authentication protocol. BAN logic uses a set of postulates to analyze the security of authentication and key agreement protocols [10], [38]. BAN logic has three elementary items, i.e., formulas/statements, principals and keys. Let X and Y be two statements, P and Q be principals, K be a symbol for a key. The basic expressions of BAN logic are described in Table 4. More details can be found in [10] and [38].

The main logical postulates of *BAN* logic are given below:

- **The message-meaning rule**:

  $$\frac{P|\equiv P\overset{K}{\longleftrightarrow}Q, P\triangleleft\{X\}_K}{P|\equiv Q|\sim X}, \frac{P|\equiv P\overset{K}{\Leftrightarrow}Q, P\triangleleft<X>_K}{P|\equiv Q|\sim X})$$

  If P believes that it shares K with Q and sees X encrypted by K (or X combined with K), then P believes that Q once said X.
- **The nonce-verification rule**: $\frac{P|\equiv\#(X), P|\equiv Q|\sim X}{P|\equiv Q|\equiv X}$
  If P believes that X could have been uttered only recently and Q once said X, then P believes that Q believes X.
- **The freshness propagation rule**: $\frac{P|\equiv\#(X)}{P|\equiv\#(X,Y)}$
  If P believes that X is fresh, then P also believes that (X, Y) is fresh.
- **The jurisdiction rule**: $\frac{P|\equiv Q|\Rightarrow X, P|\equiv Q|\equiv X}{P|\equiv X}$
  If P believes that Q has an authority on X and Q believes X, then P trusts Q on the truth of X.
- **The belief rule**: $\frac{P|\equiv Q|\equiv (X,Y)}{P|\equiv Q|\equiv X}$
  If P believes that Q believes X and Y, then P believes that Q believes X.
- **The session key rule**: $\frac{P|\equiv\#(SK), P|\equiv Q|\equiv X}{P|\equiv P\overset{SK}{\longleftrightarrow}X}$
  If P believes that the session key is fresh and P believes that Q believes X which is the necessary parameter of the session key, then P believes that P shares the session key SK with Q.

According to the analytic procedures of the BAN logic, the online authentication part of the proposed scheme must satisfy the following goals:

- **Goal (1)**: VLR $|\equiv$ (MU $\overset{SK_0}{\longleftrightarrow}$ VLR),
- **Goal (2)**: VLR $|\equiv$ MU $|\equiv$ (MU $\overset{SK_0}{\longleftrightarrow}$ VLR),
- **Goal (3)**: MU $|\equiv$ VLR $|\equiv$ (MU $\overset{SK_0}{\longleftrightarrow}$ VLR),
- **Goal (4)**: MU $|\equiv$ (MU $\overset{SK_0}{\longleftrightarrow}$ VLR).

The generic form of the online authentication part of the proposed DBAKA scheme is described as follows.

- MU$\rightarrow$VLR:

  $$\{R_1, R_2, \{R\}_{MU\overset{e(r_1P_{pub},Q_V)}{\longleftrightarrow}VLR}, e_1, e_2, \sigma', ID_V, ID_H\}.$$

TABLE 4. Notations of BAN logic.

Symbol	Descript
$P \mid\equiv X$	The principal P believes a statement X, or P would be entitled to believe X.
$P \lhd X$	P sees X. P has received a message containing X and can read and repeat X (possibly after doing some decryption).
$P \mid\sim X$	P once said X. P at some time sent a message containing X. It is not known whether this is a replay, though it is known that P believed X when he sent it.
$P \mid\Rightarrow X$	P has jurisdiction over X. P is an authority on X and is trusted on this matter.
$\#(X)$	The formula X is fresh. That is, X has never been sent in a message at any time before the current run of the protocol
$P \overset{K}{\longleftrightarrow} Q$	K is a shared key between P and Q. P and Q may use K to communicate. And K is good since it never be discovered by any principal except P or Q, or a principal trusted by either P or Q.
$P \overset{X}{\Leftrightarrow} Q$	The formula X is a shared key known only to P and Q, possibly to principals trusted by them.
$\{X\}_K$	The formula X is encrypted by K.
$<X>_Y$	This represents X combined with the formula Y. It is intended that Y be a secret and that its presence prove the identity of whoever utters $<X>_Y$. X is simply concatenated with Y while Y plays a role as proof or origin for X.
SK:	The session key used in the current session

- VLR $\rightarrow$ HLR:

$$\{e_1, e_2, \{R, R_1, R_V, ID_V, ID_H\}_{\text{VLR} \overset{k}{\longleftrightarrow} \text{HLR}}, ID_V\}.$$

- HLR$\rightarrow$VLR:

$$\{e_3, \{r_H, R_V, N_0, ID_V, ID_H, e_0\}_{\text{VLR} \overset{k}{\longleftrightarrow} \text{HLR}}\}.$$

- VLR $\rightarrow$ MU: $\{e_3, e_5\}$.

Next, we idealize the proposed DBAKA protocol in the language of formal logic as follows.

Message $\mathbf{M_1}$: MU$\rightarrow$VLR:

$$R_1, R_2, \{R\}_{\text{MU} \overset{e(r_1 P_{pub}, Q_V)}{\longleftrightarrow} \text{VLR}}.$$

Message $\mathbf{M_2}$: VLR$\rightarrow$HLR:

$$\{R, R_1, R_V, ID_V, ID_H\}_{\text{VLR} \overset{k}{\longleftrightarrow} \text{HLR}}.$$

Message $\mathbf{M_3}$: MU $\overset{via\ \text{VLR}}{\longrightarrow}$ HLR:

$$< N_0, R_1 >_{\text{MU} \overset{\sigma, R}{\longleftrightarrow} \text{HLR}}, \quad \{N_0, R, R_1\}_{\text{MU} \overset{\sigma}{\longleftrightarrow} \text{HLR}}.$$

Message M4: HLR$\rightarrow$VLR:

$$\{r_H, R_V, N_0, ID_V, ID_H, e_0\}_{\text{VLR} \overset{k}{\longleftrightarrow} \text{HLR}}.$$

Message M5: HLR $\overset{via\ \text{VLR}}{\longrightarrow}$ MU:

$$\{r_H, R_V, ID_V, ID_H, < R_1, R_V >_{\text{MU} \overset{N_0}{\longleftrightarrow} \text{HLR}}\}_{\text{MU} \overset{\sigma}{\longleftrightarrow} \text{HLR}}$$

Message M6: VLR$\rightarrow$MU:

$$\{R_1, R_V, \text{MU} \overset{\sigma}{\longleftrightarrow} \text{HLR}, \text{MU} \overset{r_H}{\Leftrightarrow} \text{HLR}, N_0\}_{\text{MU} \overset{r_1 R_V}{\longleftrightarrow} \text{VLR}}.$$

The assumptions about the initial states for the proposed DBAKA protocol are made below:

- $\boldsymbol{H_1}$: MU$\mid\equiv\#(R_1, R_2, R_3, r_H, R_V)$,
- $\boldsymbol{H_2}$: VLR$\mid\equiv\#(R_1, R_2, R_3, r_H, R_V)$,
- $\boldsymbol{H_3}$: HLR$\mid\equiv\#(R_1, r_H, R_V)$,
- $\boldsymbol{H_4}$: HLR$\mid\equiv$MU $\overset{\sigma}{\longleftrightarrow}$ HLR,
- $\boldsymbol{H_5}$: MU$\mid\equiv$MU $\overset{\sigma}{\longleftrightarrow}$ HLR,
- $\boldsymbol{H_6}$: VLR$\mid\equiv$VLR $\overset{k}{\longleftrightarrow}$ HLR,
- $\boldsymbol{H_7}$: HLR$\mid\equiv$VLR $\overset{k}{\longleftrightarrow}$ HLR,
- $\boldsymbol{H_8}$: HLR$\mid\equiv$MU$\mid\Rightarrow (N_0)$,
- $\boldsymbol{H_9}$: HLR$\mid\equiv$VLR$\mid\Rightarrow (R_V)$,
- $\boldsymbol{H_{10}}$: VLR$\mid\equiv$ VLR $\overset{e(r_1 P_{pub}, Q_V)}{\longleftrightarrow}$ MU,
- $\boldsymbol{H_{11}}$: VLR$\mid\equiv$MU$\mid\Rightarrow (R)$,
- $\boldsymbol{H_{12}}$: VLR$\mid\equiv$HLR$\Rightarrow (r_H)$,
- $\boldsymbol{H_{13}}$: VLR$\mid\equiv$VLR $\overset{r_V R_1}{\longleftrightarrow}$ MU,
- $\boldsymbol{H_{14}}$: MU$\mid\equiv$MU $\overset{r_1 R_V}{\longleftrightarrow}$ VLR,
- $\boldsymbol{H_{15}}$: MU$\mid\equiv$HLR $\overset{N_0}{\longleftrightarrow}$ MU,
- $\boldsymbol{H_{16}}$: MU$\mid\equiv$VLR$\mid\Rightarrow (R_V)$.

Now, based on the BAN logic and the assumptions, we will show that the MU and the VLR share the session key by analyzing the idealized form of the proposed DBAKA protocol.

From message M_1, we have

- $\boldsymbol{S_1}$: VLR $\lhd (R_1, R_2, \{R\}_{\text{MU} \overset{e(r_1 P_{pub}, Q_V)}{\longleftrightarrow} \text{VLR}})$.

According to S_1, H_{10}, and the message meaning rule, we have

- $\boldsymbol{S_2}$: VLR$\mid\equiv$MU$\mid\sim (R_1, R_2, R)$.

According to S_2, H_1 and the nonce-verification rule, we have

- $\boldsymbol{S_3}$: VLR$\mid\equiv$MU$\mid\equiv (R)$.

According to S_3, H_{11}, and the jurisdiction rule, we have

- $\boldsymbol{S_4}$: VLR$\mid\equiv (R)$.

From message M_2, we have

- S_5: HLR$\triangleleft\{R, R_1, R_V, ID_V, ID_H\}_{\text{VLR} \overset{k}{\longleftrightarrow} \text{HLR}}$.

According to S_5, H_7, and the message meaning, we have

- S_6: HLR $\mid \equiv$VLR$\mid \sim (R, R_1, R_V, ID_V, ID_H)$.

According to H_3 and the freshness propagation rule, we have

- S_7: HLR $\mid \equiv \#(R, R_1, R_V, ID_V, ID_H)$.

According to S_7, S_{14}, and the nonce-verification rule, we have

- S_8: HLR $\mid \equiv$VLR$\mid \equiv (R, R_1, R_V, ID_V, ID_H)$.

According to S_4, S_6, and the belief rule, we have

- S_9: HLR $\mid \equiv$VLR$\mid \equiv$ (VLR $\overset{R_V}{\longleftrightarrow}$ HLR).

According to S_9, H_9, and the jurisdiction rule, we have

- S_{10}: HLR $\mid \equiv$(VLR $\overset{R_V}{\longleftrightarrow}$ HLR).

From message M_3, we have

- S_{11}: HLR$\triangleleft\{< N_0, R_1 >_{\text{MU} \overset{\sigma,R}{\longleftrightarrow} \text{HLR}}, \{N_0, R, R_1\}_{\text{MU} \overset{\sigma}{\longleftrightarrow} \text{HLR}}\}$.

According to S_{11}, H_4, and the message meaning, we have

- S_{12}: HLR $\mid \equiv$MU$\mid \sim (N_0, R_1)$.

According to H_3 and the freshness propagation rule, we have

- S_{13}: HLR $\mid \equiv \#(N_0, R_1)$.

According to S_{12}, S_{13}, and the nonce-verification rule, we have

- S_{14}: HLR $\mid \equiv$MU$\mid \equiv (N_0, R_1)$.

According to S_{14} and the belief rule, we have

- S_{15}: HLR $\mid \equiv$MU$\mid \equiv$(MU$\overset{N_0}{\longleftrightarrow}$ HLR).

According to S_{15}, H_8, and the jurisdiction rule, we have

- S_{16}: HLR $\mid \equiv$(MU$\overset{N_0}{\longleftrightarrow}$ HLR).

From message M_4, we have

- S_{17}:VLR$\triangleleft\{r_H, R_V, N_0, ID_V, ID_H, e_0\}_{\text{VLR} \overset{k}{\longleftrightarrow} \text{HLR}}$.

According to S_{17}, H_6, and the message meaning, we have

- S_{18}: VLR $\mid \equiv$ HLR $\mid \sim (r_H, R_V, N_0, ID_V, ID_H, e_0)$.

According to H_2 and the freshness propagation rule, we have

- S_{19}: VLR $\mid \equiv \#(r_H, R_V, N_0, ID_V, ID_H, e_0)$.

According to S_{18}, S_{19}, and the nonce-verification rule, we have

- S_{20}: VLR $\mid \equiv$HLR$\mid \equiv (r_H, R_V, N_0, ID_V, ID_H, e_0)$.

According to S_{20} and the belief rule, we have

- S_{21}: VLR $\mid \equiv$HLR$\mid \equiv$(VLR $\overset{r_H, N_0, e_0}{\longleftrightarrow}$ HLR).

According to S_{10}, S_{16}, S_{21}, H_{12}, and the jurisdiction rule, we have

- S_{22}: VLR $\mid \equiv (r_H, N_0, e_0)$.

According to S_{22}, H_{13}, and the session key rule, we have

- S_{23}: VLR$\mid \equiv$(MU $\overset{SK_0}{\longleftrightarrow}$VLR). (**Goal** (1))

According to S_{23}, H_2, and the nonce-verification rule, we have

- S_{24}:VLR$\mid \equiv$MU$\mid \equiv$(MU $\overset{SK_0}{\longleftrightarrow}$VLR). (**Goal** (2))

From message M_5, we have

- S_{25}: MU $\triangleleft \{r_H, R_V, ID_V, ID_H, < R_1, R_V >_{\text{MU} \overset{N_0}{\longleftrightarrow} \text{HLR}}\}_{\text{MU} \overset{\sigma}{\longleftrightarrow} \text{HLR}}$.

According to S_{25}, H_5, and the message meaning, we have

- S_{26}: MU$\mid \equiv$VLR$\mid \sim (r_H, R_V, ID_V, < R_1, R_V, \sigma >_{N_0})$.

According to H_1 and the freshness propagation rule, we have

- S_{27}: MU$\mid \equiv \#(r_H, R_V, ID_V, < R_1, R_V, \sigma >_{N_0})$.

According to S_{26}, S_{27}, and the nonce-verification rule, we have

- S_{28}: MU$\mid \equiv$HLR$\mid \equiv (r_H, R_V, N_0, ID_V, ID_H, e_0)$.

According to S_{28}, H_{15}, and the belief rule and the message meaning rule, we have

- S_{29}: MU$\mid \equiv (r_H, R_V)$.

From message M_6, we have

- S_{30}: MU$\triangleleft\{R_1, R_V, \text{MU} \overset{\sigma}{\longleftrightarrow} \text{HLR}, \text{MU} \overset{r_H}{\Leftrightarrow} \text{HLR}, N_0\}_{\text{MU} \overset{\mathrm{r}_1 R_V}{\longleftrightarrow} \text{VLR}}$.

According to S_{30}, H_{14}, and the message meaning, we have

- S_{31}: MU$\mid \equiv$VLR$\mid \sim\{R_1, R_V, \text{MU} \overset{\sigma}{\longleftrightarrow} \text{HLR}, \text{MU} \overset{r_H}{\Leftrightarrow} \text{HLR}, N_0\}$.

According to H_1 and the freshness propagation rule, we have

- S_{32}: MU $\mid \equiv \#(R_1, R_V, \text{MU} \overset{\sigma}{\longleftrightarrow} \text{HLR}, \text{MU} \overset{r_H}{\Leftrightarrow} \text{HLR}, N_0)$.

According to S_{31}, S_{32}, and the nonce-verification rule, we have

- S_{33}: MU$\mid \equiv$VLR$\mid \equiv (R_1, R_V, \text{MU} \overset{\sigma}{\longleftrightarrow} \text{HLR}, \text{MU} \overset{r_H}{\Leftrightarrow} \text{HLR}, N_0)$.

According to S_{29}, S_{33}, H_{14}, and the belief rule, since $e_0 = h_3(r_H \| \sigma \| ID_V \| ID_H)$,$SK_0 = h_6(N_0 \| r_H \| e_0 h_4(r_1 R_V)))$, we have

- S_{34}: MU$\mid \equiv$VLR$\mid \equiv$(MU$\overset{SK_0}{\longleftrightarrow}$ VLR). (**Goal** (3))

According to S_{29}, H_{16}, and the jurisdiction rule, we have

- S_{35}: MU$\mid \equiv$ (MU$\overset{SK_0}{\longleftrightarrow}$ VLR). (**Goal** (4))

Next, we prove the offline authentication protocol secure. Based on BAN logic, we will show that the offline authentication protocol must satisfy the following goals.

- **Goal (5)**: VLR $\mid \equiv$ (MU $\overset{SK_i}{\longleftrightarrow}$ VLR),
- **Goal (6)**: VLR$\mid \equiv$MU $\mid \equiv$ (MU$\overset{SK_i}{\longleftrightarrow}$VLR),
- **Goal (7)**: MU $\mid \equiv$ VLR$\mid \equiv$(MU$\overset{SK_i}{\longleftrightarrow}$VLR),
- **Goal (8)**: MU $\mid \equiv$ (MU$\overset{SK_i}{\longleftrightarrow}$VLR).

First, the offline authentication protocol is described into idealized form below.

- Message M_7: MU$\rightarrow$VLR:

$$\{< N_i >_{N_{i-1}}, A_i, < N_i, A >_{r_H}\}_{SK_{i-1}}.$$

- Message M_8: VLR$\rightarrow$MU:

$$\{B_i, \text{MU} \overset{N_i}{\longleftrightarrow} \text{VLR}, \quad \text{MU} \overset{r_H}{\Leftrightarrow} \text{VLR}, < SK_i >_{\text{MU} \overset{b_i A_i}{\longleftrightarrow} \text{VLR}}\}_{\text{MU} \overset{SK_{i-1}}{\longleftrightarrow} \text{VLR}}.$$

Second, we make the assumptions about the initial state of the offline authentication protocol as follows.

- $\boldsymbol{H_{17}}$: MU| $\equiv \#(A_i, B_i)$,
- $\boldsymbol{H_{18}}$: VLR| $\equiv \#(A_i, B_i)$,
- $\boldsymbol{H_{19}}$: MU | $\equiv$VLR | $\equiv$ (B_i),
- $\boldsymbol{H_{20}}$: MU| $\equiv$MU$\overset{SK_{i-1}}{\longleftrightarrow}$HLR,
- $\boldsymbol{H_{21}}$: VLR | $\equiv$ MU $\overset{SK_{i-1}}{\longleftrightarrow}$VLR,
- $\boldsymbol{H_{22}}$: VLR | $\equiv$ MU$\overset{N_{i-1}}{\longleftrightarrow}$VLR,
- $\boldsymbol{H_{23}}$: VLR | $\equiv$ MU | $\equiv$ (N_i),
- $\boldsymbol{H_{24}}$: VLR | $\equiv$MU$\overset{b_iA_i}{\longleftrightarrow}$VLR,
- $\boldsymbol{H_{25}}$: MU| $\equiv$MU$\overset{a_iB_i}{\longleftrightarrow}$VLR.

Third, we analyze the idealized form of the proposed offline authentication protocol based on the BAN logic and the assumptions. The main proofs are stated below.

From message M_7, we have

- $\boldsymbol{S_{36}}$: VLR$\triangleleft\{< N_i >_{N_{i-1}}, A_i, < N_i, A >_{r_H}\}_{SK_{i-1}}$.

According to S_{36}, H_{21}, and the message meaning, we have

- $\boldsymbol{S_{37}}$: VLR | $\equiv$MU| $\sim \{< N_i >_{N_{i-1}}, A_i, < N_i, A_i >_{r_H}\}$.

According to H_{18} and the freshness propagation rule, we have

- $\boldsymbol{S_{38}}$: VLR | $\equiv \#(< N_i >_{N_{i-1}}, A_i, < N_i, A_i >_{r_H})$.

According to S_{37}, S_{38}, and the nonce-verification rule, we have

- $\boldsymbol{S_{39}}$: VLR| $\equiv$MU| $\equiv$ $(< N_i >_{N_{i-1}}, A_i)$.

According to S_{39} and the belief rule, we have

- $\boldsymbol{S_{40}}$: VLR| $\equiv$MU| $\equiv$ $(< N_i >_{N_{i-1}})$.
- $\boldsymbol{S_{41}}$: VLR| $\equiv$MU| $\equiv$ (A_i).

According to S_{40}, H_{22}, and the message meaning rule and the nonce verification rule, we have

- $\boldsymbol{S_{42}}$: VLR| $\equiv$MU| $\equiv$ (N_i).

According to S_{42}, H_{23}, and the jurisdiction rule, we have

- $\boldsymbol{S_{43}}$: VLR | $\equiv$ (N_i).

According to S_{41}, S_{43}, H_{24}, and the nonce-verification rule, we have

- $\boldsymbol{S_{44}}$: VLR | $\equiv$ (MU$\overset{SK_i}{\longleftrightarrow}$ VLR). (**Goal** (5))

According to S_{44}, H_{18}, and the session key rule, we have

- $\boldsymbol{S_{45}}$: VLR | $\equiv$ MU| $\equiv$ (MU$\overset{SK_i}{\longleftrightarrow}$ VLR). (**Goal** (6))

From message M_8, we have

- $\boldsymbol{S_{46}}$: MU$\triangleleft\{B_i$, MU$\overset{N_i}{\longleftrightarrow}$ VLR,

MU $\overset{r_H}{\Leftrightarrow}$ VLR, $< SK_i >_{\text{MU} \overset{b_iA_i}{\longleftrightarrow} \text{VLR}}\}_{\text{MU} \overset{SK_{i-1}}{\longleftrightarrow} \text{VLR}}$.

According to S_{46}, H_{20}, and the message meaning, we have

- $\boldsymbol{S_{47}}$: MU | $\equiv$ VLR | $\sim\{B_i$, MU$\overset{N_i}{\longleftrightarrow}$ VLR,

MU $\overset{r_H}{\Leftrightarrow}$ VLR, $< SK_i >_{\text{MU} \overset{b_iA_i}{\longleftrightarrow} \text{VLR}}\}$.

According to H_{17} and the freshness propagation rule, we have

- $\boldsymbol{S_{48}}$: MU| $\equiv \#(B_i$, MU$\overset{N_i}{\longleftrightarrow}$ VLR, MU$\overset{r_H}{\Leftrightarrow}$ VLR, $< SK_i >_{\text{MU} \overset{b_iA_i}{\longleftrightarrow} \text{VLR}})$.

According to S_{46}, S_{48}, and the nonce-verification rule, we have

- $\boldsymbol{S_{49}}$: MU | $\equiv$ VLR| $\equiv$ $(B_i$, MU$\overset{N_i}{\longleftrightarrow}$ VLR, MU $\overset{r_H}{\Leftrightarrow}$ VLR, $< SK_i >_{\text{MU} \overset{b_iA_i}{\longleftrightarrow} \text{VLR}}$).

According to S_{49}, H_{20}, H_{25}, and the belief rule, we have

- $\boldsymbol{S_{50}}$: MU| $\equiv$VLR| $\equiv$ (MU$\overset{SK_i}{\longleftrightarrow}$ VLR). (**Goal** (7))

According to S_{43}, S_{50}, H_{19}, and the jurisdiction rule, we have

- $\boldsymbol{S_{51}}$: MU | $\equiv$ (MU$\overset{SK_i}{\longleftrightarrow}$ VLR). (**Goal** (8))

The above proofs demonstrate that the MU, the VLR and the HLR achieve mutual authentication property, and the MU and the VLR establish session keys securely.

C. DISCUSSION ON OTHER ATTACKS

According to the adversary model mentioned earlier, we will give a detailed security analysis of the proposed DBAKA scheme. The informal security analysis shows that our scheme is also secure against the following known attacks.

1) DOS ATTACKS (S1)

During the online authentication phase, the proxy key (R, σ) of MU and the database of HLR do not require synchronous updates. Next, for each authentication request, HLR first validates it by decrypting the message. Each online authentication only requires HLR to choose a random number r_H.

During the offline authentication phase, MU extracts (N_i, SK_{i-1}, r_H) from the SIM card and sends the ciphertext f_i to VLR. VLR verifies the login message by checking if $h(N_i)$ is the same as N_{i-1} and the hash value is equal to $h((N_i \oplus r_H)\|A_i)$. If they both hold, the VLR replaces N_i with N_{i-1} and computes the session key. Thus, the VLR can control the amount of incoming login messages during the offline authentication phase. Therefore, any adversary is not able to disturb the availability of the authentication between a legitimate MU and VLR.

2) REQUEST REPLICATION ATTACKS (S2)

In order to mount request replication attack, a malicious visited location register (as an adversary) generally transmits another VLR's messages of previous protocol run to the HLR and attempts to prove that the message is sent from a legal visited location register. Assume that a malicious VLR' intercepts MU's login message $\{R_1, R_2, R_3, e_1, e_2, \sigma', ID_V, ID_H\}$ to VLR. If VLR' attempts to replace $\{e_V, ID_V\}$ with a new message $(e_{V'}, ID_{V'})$, since $e_V = E_k(R\|R_1\|R_V\|ID_V\|ID_H)$, VLR' has to obtain R. Since VLR' is not able to compute k, the adversary cannot recover R. Next, HLR can also figure out that VLR' is invalid by decrypting e_1 and then verifying e_2. This is because that e_2 is the hash value of R, N_0, ID_V, etc. Thus, VLR' fails in mounting the request replication attacks.

3) IMPERSONATION ATTACKS (S3)

In our system model, assume that an adversary has eavesdropped on all the messages transmitted over the public

channel during the protocol execution. Moreover, the adversary can modify and re-transmit the messages.

First, we will illustrate that the proposed protocol can resist the user impersonation attacks. Assume that an adversary has trapped all the access request messages $\{R_1, R_2, R_3, e_1, e_2, \sigma', ID_V, ID_H\}$ during the online authentication phase. Then the adversary attempts to generate another valid message. The adversary must compute a valid signature σ' on message $\{R_1, R_2, R_3, e_1, e_2, ID_V, ID_H\}$. Since Schnorr signature is proven unforgeable under chosen message attacks in the Random Oracle model [39], it is infeasible for any adversary to generate a valid signature σ' without the key (R, σ). Moreover, due to the unknown key (R, σ), it is infeasible to generate a valid pair $\{e_1, e_2\}$ within polynomial time. Similarly, assume that an adversary has trapped the access request message f_i during the offline authentication phase. It is clear that the adversary is not able to compute another valid f_j without the knowledge of $\{SK_{j-1}, N_{j-1}, N_j, r_H\}$.

Next, we will demonstrate that the proposed protocol resists the VLR impersonation attacks. Assume that an adversary has intercepted the messages $\{e_1, e_2, e_V, ID_V\}/\{e_3, e_5\}$ during the online authentication phase. Then the adversary tries to generate message $\{e_1, e_2, e'_V, ID_V\}$ and $\{e_3, e'_5\}$ and transmits them to HLR and MU, respectively. If HLR or MU accepts the message $\{e_1, e_2, e'_V, ID_V\}$ or $\{e_3, e'_5\}$, the adversary will succeed in impersonating as a valid VLR. However, the computation of e'_V relies on the secret key k and the shared parameter R, while e'_5 depends on the secret parameters $\{r_H, N_0, e_0\}$. Due to the Difficulty of Computational Diffie-Hellman Problem and one-wayness of cryptographic hash function, the adversary cannot extract the shared secret key k from public messages. The adversary guesses valid secret parameters $\{r_H, N_0, e_0\}$ with the probability $1/q^3$. Assume that an adversary has intercepted the message $\{f_i, g_i\}$ during the offline authentication phase. Then the adversary tries to generate another message g_j and transmits it to MU. It is obvious that the adversary is not able to compute another valid g_j without the knowledge of $\{SK_{j-1}, N_{j-1}, N_j, r_H\}$.

Finally, we show that the proposed protocol resists the HLR impersonation attacks. Assume that an adversary has intercepted the message $\{e_3, e_4\}$. Then the adversary tries to generate another message $\{e'_3, e'_4\}$ and transmits it to VLR. Note that e'_3 is decided by the proxy key σ and the shared parameters $\{r_H, N_0, R_V\}$, while e'_4 is computed from the proxy key k and the shared secret parameters $\{r_H, N_0, e_0\}$. Due to the hardness assumption of ECDH Problems, the adversary cannot extract the shared secret keys k from public messages. Due to the non-invertible property of hash function, the adversary cannot compute the shared secret parameters $\{r_H, N_0, e_0\}$ from the intercepted messages. Therefore, the adversary generates valid secret parameters $\{r_H, N_0, e_0\}$ with the probability $1/q^3$.

The above analysis shows that the adversary is not able to launch impersonation attacks within polynomial time.

4) KNOWN-KEY ATTACKS (S4)

To evaluate the damage of leakage of a set of consecutive session keys and the long-term keys, we discuss the probability of deriving a session key from the compromised secret keys and the compromised session keys. In the setting of the delegation-based authentication protocol, we assume that the proxy key pair (R, σ) of the user MU, the secret key S_H of the HLR, the secret key S_V of the VLR, and even the system secret parameter s are disclosed to the adversary and the adversary tries to generate a session key from some compromised session keys. According to the origin of the compromised session keys, we discuss this issue in the two cases: online authentication and offline authentication.

Case 1 (Online Authentication): Assume that an adversary has known all the participants' long-term secret keys and a set of m consecutive session keys $SK_0^{(i-1)}, SK_0^{(i-2)}, \ldots, SK_0^{(i-m)}$ of the $(i-1)$-th, $(i-2)$-th,...$(i\text{-}m)$-th online authentication and tries to generate a session key $SK_0^{(i)}/SK_0^{(i-m-1)}$ of the i-th/$(i\text{-}m\text{-}1)$-th online authentication. Further assume that the adversary has already intercepted all the messages $\{ID_V, ID_H, R_1^{(j)}, R_2^{(j)}, R_3^{(j)}, e_1^{(j)}, e_2^{(j)}, e_3^{(j)}, e_4^{(j)}, e_5^{(j)}, e_V^{(j)}, \sigma'^{(j)}\}$ $(j = i, i-1, i-2, \ldots, i\text{-}m$ *or* $i\text{-}1, i\text{-}2, \ldots, i\text{-}m\text{-}1)$ transmitted among MU, VLR and HLR. With the proxy key σ and k, the adversary decrypts $\{e_1^{(j)}, e_V^{(j)}, e_3^{(j)}, e_4^{(j)}\}$ and extracts the random factors $\{N_0^{(j)}, r_H^{(j)}, R_V^{(j)}\}$ $(j = i, i-1, i-2, \ldots, i\text{-}m$ *or* $i\text{-}1, i\text{-}2, \ldots, i\text{-}m\text{-}1)$. However, the adversary still fails in computing the session key $SK_0^{(i)}/SK_0^{(i-m-1)}$. The adversary has to compute $h_4(r_V^{(i)}R_1^{(i)})$ or $h_4(r_V^{(i-m-1)}R_1^{(i-m-1)})$. This is because $SK_0^{(l)} = h_6(N_0^{(l)} \| r_H^{(l)} \| e_0^{(l)} \| h_4(r_V^{(l)}R_1^{(l)}))$, $l = i$ or $i\text{-}m\text{-}1$, It is computationally infeasible for the adversary to compute $r_1^{(i-1)}R_V^{(i-1)}$ from the pair $(R_1^{(i-1)}, R_V^{(i-1)})$ without $r_1^{(i-1)}$ or $r_V^{(i-1)}$ due to the hardness assumption of ECDH Problems. In this way, the proposed protocol provides the secrecy of session keys.

Case 2 (Offline Authentication): Assume that an adversary has known all the participants' long-term secret keys and a set of m consecutive session keys $SK_{i-1}, SK_{i-2}, \ldots, SK_{i-m}$ of the $(i\text{-}1)$-th, $(i\text{-}2)$-th,... $(i\text{-}m)$-the offline authentication and attempts to compute the session key SK_i/SK_{i-m-1} of the i-th/$(i\text{-}m\text{-}1)$-th offline authentication. Further assume that the adversary has intercepted all the messages $\{f_j, g_j\}$ $(j = i, i-1, i-2, \ldots, i\text{-}m$ *or* $i\text{-}1, i\text{-}2, \ldots, i\text{-}m\text{-}1)$ transmitted between MU and VLR. With the session key SK_{i-1}, the adversary decrypts $\{f_i, g_i\}$ to obtain $\{A_i, B_i\}$ and extract the number N_i with the known N_{i-1}. Due to the hardness assumption of ECDH Problems, it is computationally infeasible for the adversary to compute b_iA_i or a_iB_i from the pair (A_i, B_i). Thus, the adversary cannot compute the session key SK_i. Similarly, with the session key SK_{i-m}, the adversary decrypts $\{f_{i-m}, g_{i-m}\}$ and extracts N_{i-m-1}. However, the adversary is not able to decrypt $\{f_{i-m-1}, g_{i-m-1}\}$. Thus, the adversary cannot obtain $\{A_{i-m-1}, B_{i-m-1}\}$. Although the adversary has N_{i-m-1} and r_H, it cannot still recover the session key $SK_{i-m-1} = h_6(N_{i-m-1} \| SK_{i-m-1} \| h(b_{i-m-1}A_{i-m-1}) \| r_H)$.

Therefore, the proposed protocol eventually achieves known key secrecy.

D. FUNCTIONALITY ANALYSIS

In this section, we show that the proposed scheme can fulfill the following functional requirements.

1) MUTUAL AUTHENTICATION (F1)

During the online authentication phase, VLR authenticates MU by checking if the verification equation of the proxy signature holds in Step 4, while VLR authenticates HLR by checking if the plaintext R'_V is equal to R_V, and ID_V and ID_H are valid identities. MU authenticates VLR and HLR by checking if the hash value of $\{\sigma, R_1, R_V, N'_0\}$ is equal to $h_3(\sigma \| R_1 \| R_V \| N_0)$ and the decrypted strings $\{ID_H, ID_V\}$ are equal to HLR's identity and VLR's identity, respectively. Here $\{\sigma, R_1, R_V, N'_0\}$ is derived from the ciphertext e_3. Without knowledge of the secret proxy key σ, it is impossible to generate a valid e_3 and impersonate HLR. HLR authenticates VLR by checking whether ID'_V is the same as the received identity and ID'_H is its identity where ID'_V and ID'_H are from the ciphertext e_V under the key k. Based on the message $\{e_1, e_2\}$, HLR authenticates MU. After HLR decrypts e_1 with the proxy key σ which corresponds with R, HLR validates e_2.

During the offline authentication phase, upon receiving the login message f_i from MU, VLR checks whether $h_1(N_i) = N_{i-1}$ and $h_1((N_i \oplus r_H)\|A_i)$ is valid. If they are valid, VLR has successfully authenticated the MU. It is infeasible that any adversary generates the login message to pass the verification of VLR, since the message is encrypted with the previous session key SK_{i-1}. Similarly, MU authenticates VLR by checking the validity of the hash value $h_5(h_6(N_i\|SK_{i-1}\|h_4(b_iA_i)\|r_H)\|B_i)$ which is derived from g_i.

Therefore, the proposed protocol negotiates the session key after performing mutual authentication.

2) NON-REPUDIATION (F2)

At the first stage of the proposed protocol execution, MU precomputes a hash chain $h_1^{(1)}(r_M), h_1^{(2)}(r_M), \ldots, h_1^{(n+1)}(r_M)$, i.e. $N_0, N_1, \ldots, N_n$. During the online authentication phase, MU signs $h_1^{(n+1)}(r_M)$ (it is hashed into e_2) with the proxy key pair. Then MU sends the proxy signature σ' to VLR. The proxy key (R, σ) is authorized by HLR. Since the proxy key and the proxy signature are generated from Schnorr signature, while Schnorr signature is proven unforgeable under chosen message attacks in the Random Oracle model [39], it is infeasible for any adversary to generate the proxy key pair and the signature. Thus, if the proxy signature is valid, VLR is sure that the MU is a legal user of HLR. Because only HLR can authorize MU to sign on his behalf, once any dispute happens, neither HLR nor MU can deny the access request and authentic message (i.e., proxy signature).

During the offline authentication phase, MU obtains $\{N_i, r_H\}$ from the SIM card, computes $f_i = E_{SK_{i-1}}(N_i \oplus N_{i-1}, A_i, h_1((N_i \oplus r_H)\|A_i))$ and sends f_i to VLR. When VLR receives these messages from MU, VLR recovers N_i by using the session key SK_{i-1} and r_H. Then VLR checks whether $h(N_i)$ and the hash value are the same as the stored value N_{i-1} and $h_1((N_i \oplus r_H)\|A_i)$, respectively. If they are the same, VLR authenticates the MU. Only a legal MU can compute f_i and the session key SK_i. In essence, any adversary could not masquerade MU to compute the authenticated ciphertext f_i and to deceive the VLR. Thus, even if the dispute occurs during the offline authentication phase, MU cannot deny the fact that he/she has gained access to VLR.

3) WEAK UN-TRACEABILITY (F3)

User privacy is an important issue in the roaming scenario. We will show that the proposed protocol hides the roaming user's identity from any eavesdroppers and the foreign servers. Moreover the proposed protocol hides user's movements from any eavesdroppers and other foreign servers (except the VLR).

The proposed protocol uses the proxy key (R, σ) and the previous session key instead of any real identity of MU to execute online authentication and offline authentication, respectively. σ is kept secret by the MU and HLR. R is not transmitted in the form of plaintext over the public network during the online authentication phase. Without knowledge of the VLR's private key S_V, it is impossible to work R out from R_3 since R is contained in R_3. During the online authentication phase, the message $\{R_1, R_2, R_3, e_1, e_2, \sigma', ID_V, ID_H\}$ sent by the MU is generated with random numbers r_1 and r_2. It will change with each run of the protocol. Furthermore, the message e_V is randomized by random numbers R_1 and R_V. Thus, any adversary cannot trace the MU.

Besides, during the offline authentication phase, MU sends different ciphertexts on different hash values with the previous session key. The adversary cannot use old ciphertexts to figure out the trace of MU.

Therefore, besides MU and HLR, no one including VLR is able to identify MU. Furthermore, other VLRs cannot decide if MU has involved any protocol runs. The proposed protocol can achieve the user weak un-traceability.

4) COMMUNICATION CONFIDENTIALITY (F4)

During the online authentication phase, after performing mutual authentication, the protocol has agreed upon some common secret parameters $\{r_H, N_0, R_V\}$ and a public parameter R_1 among all the entities MU, VLR and HLR. However, due to the Diffie-Hellman exchange, only MU and VLR can compute the shared hash $h_4(r_1R_V)$ or $h_4(r_VR_1)$. Upon the hardness assumption of the discrete logarithm problem, HLR is not able to calculate $h_4(r_1R_V)$. Thus, it is impossible for HLR to compute the session key SK_0. This is since $SK_0 = h_6(N_0\|r_H\|e_0\|h_4(r_VR_1))$.

During the offline authentication phase, after performing mutual authentication, the MU and the VLR have exchanged A_i and B_i with the shared previous session key SK_{i-1}. Through the similar analysis of the online case, since $SK_i = h_6(N_i\|SK_{i-1}\|h_4(b_iA_i)\|r_H)$, one can infer that only the MU and VLR are able to compute a Diffie-Hellman-like session key SK_i.

Note that the session key in the online case is sent through the public channel between MU and VLR. However, it is protected by the one-way function. While the session key in the offline case is also transmitted through the public channel between MU and VLR, it first is hashed and then the hash value is encrypted. Therefore, any adversary even including HLR is not able to work out the session key from it. The protocol provides strong security protection on the session key against any adversary.

VI. SECURITY, FUNCTIONALITY AND PERFORMANCE COMPARISON

In this section, we will make comparison with the related DBAKA protocols for wireless roaming service in terms of security, functionality and performance.

A. SECURITY COMPARISON

As shown as in Section V, the proposed scheme provides key agreement, mutual authentication, and user weak untraceability. Furthermore, the scheme also provides communication confidentiality. The proposed scheme is also resilient against denial of service attacks, request replication attacks and impersonation attacks. It has removed the vulnerability of the DBAKA protocols for wireless roaming service [8], [10], i.e. known key attacks.

We compare security of the proposed scheme with existing DBAKA protocols [4], [8], [10], [11], [30], [32], [33], [35]. A detailed security comparison is tabulated in Table 5. Table 5 shows that the proposed scheme overcomes most of the security weaknesses of the existing related schemes.

TABLE 5. Security comparison.

	S1	*S2*	*S3*	*S4*
[4]	×	√	√	×
[30]	×	√	√	×
[32]	×	×	√	×
[33]	√	×	√	×
[35]	√	√	√	×
[11]	√	√	√	×
[8]	√	√	√	×
[10]	√	√	√	×
Ours	√	√	√	√

Note: S1: denial-of-service attack, S2: request replication attack, S3: impersonation attack, S4: known-key attack. ×: insecure against a particular attack, √: secure against a particular attack.

B. FUNCTIONALITY COMPARISON

In Table 6, we compare different functionalities of our proposed scheme with the existing related DBAKA protocols [4], [8], [10], [11], [30], [32], [33], [35]. The tabulated result shows that none of the existing schemes provide communication confidentiality. These DBAKA protocols for wireless roaming service also require the certificate based public key of HLR. Since VLR also always acts as a HLR, the management of the keys for HLRs or VLRs is not simple. In contrast,

TABLE 6. Functionality comparison.

	F1	*F2*	*F3*	*F4*	*F5*	*F6*	*F7*
[4]	×	×	×	×	×	×	×
[30]	√	√	×	×	×	×	×
[32]	√	√	×	×	×	×	×
[33]	√	√	×	×	×	×	×
[35]	√	√	×	×	×	×	×
[11]	√	√	×	×	×	×	×
[8]	√	√	×	×	×	×	×
[10]	√	√	×	×	×	×	√
Ours	√	√	√	√	√	√	√

Note: F1: mutual authentication, F2: non-repudiation, F3: weak untraceability, F4: communication confidentiality, F5: key confirmation, F6: no certificate based public key, F7: no secrets pre-sharing. ×: does not hold a particular feature, √: holds a particular feature.

our DBAKA protocol for wireless roaming service is based on identity-based cryptography. Its key management is nearly as simple as that of GSM. Furthermore, the existing related DBAKA protocols for wireless roaming service require that VLR and HLR must share secrets K_{HV} in advance. It is inconvenient for the VLR and HLR. Since there are many VLRs for a HLR in the portable communication system, each HLR has to share a secret with every VLR. HLR also always acts as a VLR. Thus, HLR must store some more shared secrets again as a VLR. As the number of HLRs and VLRs enlarges, the number of secrets stored by HLR and VLR will increase rapidly. Our DBAKA protocol for wireless roaming service does not require VLR and HLR to share any secret key in advance. When they need to transmit message to each other, they compute the shared secret from each other's identity.

C. PERFORMANCE COMPARISON

Since Tsai et al.'s DBAKA protocol for wireless roaming service [11] is efficient and Hwang et al.'s DBAKA protocol for wireless roaming service [10] has stronger security, we only give the comparison results among the proposed scheme and the two schemes in term of storage cost, computational cost and communication cost.

TABLE 7. Storage required at the MU side.

	Tsai et al.[11]	Hwang et al.[10]	Our
Size (in bits)	1344+128n	6120+128n	1856+128n

Assume that the order q of the generator P in the elliptic curve group G_1 is a 160-bit prime and hash values are 128-bit strings (e.g. use MD5). Consider that the authentication for TMISs consists of the VLR, the HLR and a resource constrained mobile device. We only discuss the storage requirements of MU. In Table 7, we tabulate the storage size at the MU side. The proposed DBAKA protocol requires MU's device to store more 512 bits than Tsai et al.'s DBAKA protocol for wireless roaming service. However, it is

TABLE 8. Computation costs.

		Tsai et al.[11]	Hwang et al.[10]	Our
For online authentication	MU	$1T_h+2T_s+1t_m+1t_{sym}+2t_h$	$8t_e+8t_h+1t_m+2T_M+2t_{sym}+1T_{sig}$	$2T_p+2T_e+4T_s+2T_m+1T_a+1t_{sym}+7t_h$
	VLR	$1T_h+4T_s+2t_{sym}+2t_h+2T_a$	$2t_e+4t_h+1T_M+2t_{sym}+1T_{vsig}$	$4T_p+1T_e+1T_{inv}+2T_s+4T_m+1t_m+2t_{sym}+4t_h$
	HLR	$3t_{sym}+1t_h$	$7t_e+9t_h+2T_M+3t_{symv}+1t_m+1t_{inv}$	$1T_p+3t_{sym}+3t_h$
For offline authentication	MU	$2T_h+1t_{sym}$	$4T_h+2t_{sym}$	$4T_h+2t_{sym}+2T_s$
	VLR	$2T_h+1t_{sym}$	$4T_h+2t_{sym}$	$4T_h+2t_{sym}+2T_s+1t_m$

practically insignificant considering that fact that the most current mobile devices, including 4G cellular phones, personal digital assistants (PDAs) and notebook computers, has over a few hundred MB or a few GB of available memory. In contrast to our protocol, Hwang et al.'s DBAKA protocol for wireless roaming service requires the storage of $(6120+128n)$-bit strings at the MU side.

A DBAKA protocol for wireless roaming service consists of phases: setup, registration, online authentication and offline authentication. Since the setup and registration phases have been completed before the session key agreement begins, we only tabulate the computational costs at the user side and the server side for the online/offline authentication. We ignore lightweight computations, such as the XOR operations and the addition operation in Z_q.

By T_p, T_s, T_m, T_e, T_a, T_{inv}, T_h, t_{sym}, t_h, t_m, T_M, t_e, t_{inv}, T_{sig}, and T_{vsig}, we denote the time required to perform one pairing operation, one scalar multiplication in G_1, one multiplication in G_2, one exponentiation in G_2, one point addition operation in G_1, one inverse operation in G_2, one map-to-point hash operation, one symmetric encryption/decryption operation, one hash operation, one modular multiplication in Z_q, one multiplication, one exponentiation, one inverse operation in a field, one signature generation and one signature verification, respectively.

In Table 8, we analyze the efficiency on computation costs of the proposed scheme and the schemes [10], [11]. According to [40]–[42], the time of different operations satisfies the following: $T_p \approx 1440t_m$, $T_s \approx 29t_m$, $T_e \approx 21t_m$, $T_h \approx 23t_m$, $T_m \approx t_h \approx t_m$, $t_{sym} \approx 3t_m$, $T_a \approx 0.12t_m$, and $T_{inv} \approx 240t_m$. It can be observed that the proposed scheme is more computationally costly which in contrary is a result of providing enhanced security and more functionality properties with respect to the related DBAKA protocols for wireless roaming service [10], [11] as shown in Tables 5 and 6.

Now we evaluate the computing time of the operations. Compared with other operations, bilinear pairing computation is more expensive. Fortunately, the detailed simulation shows [22] that by using the Pairing-Based Cryptography (PBC) library [43] and the GNU Multiple Precision (GMP) arithmetic library [44], one bilinear pairing operation requires 74.1ms, 2.9ms for iPhone 4S and Intel Core i5-2500 (at 3.30 GHz), respectively. During online authentication, at the side of mobile devices, the computational costs of the proposed scheme, Tsai et al.'s scheme and Hwang et al.'s scheme are 156.95, 20.48, and 528.28 milliseconds, respectively. At the side of VLR(HLR), the computational costs of the proposed scheme, Tsai et al.'s scheme and Hwang et al.'s scheme are 44.27(2.92), 26.45(2.02), and 278.82(360.82) milliseconds, respectively. During offline authentication, at the side of mobile devices, the computational costs of the proposed scheme, Tsai et al.'s scheme and Hwang et al.'s scheme are 3.5, 2.51, and 5.26 milliseconds, respectively. At the side of VLR, the computational costs of the proposed scheme, Tsai et al.'s scheme and Hwang et al.'s scheme are 15.14, 1.12, and 22.01 milliseconds, respectively. The proposed scheme performs better than the protocol in [11]. The protocol in [11] has better performance than the proposed scheme. However, the scheme fails to achieve weak un-traceablility, communication confidentially and resist against known key attacks. Hence, considering the higher security level and the more functionality properties, the computational overhead of the proposed scheme in comparison to Tsai et al.'s DBAKA protocol is acceptable.

TABLE 9. Communication costs.

Total bits required	[11]	[10]	Our
For online authentication	4384	$18720+2\|m_M\|$	8928
For offline authentication	128	416	768

Note: $|m_M|$ is the size of the registration document about private/public key.

In Table 9, we tabulate the number of bits required for each communication among the proposed scheme and other related existing schemes. We assume that bit size of the identity and random numbers are 160 bits and 128 bits, respectively. The block size (for example, AES-128) of symmetric encryption/decryption is 128 bits. Since registration is executed only once, we concentrate on the message exchange during the login and authentication phases. The proposed scheme requires less communication cost as compared to that for Hwang et al.'s scheme. Though the proposed scheme requires more communication cost as compared to that for Tsai et al.'s scheme, it provides various security and functionality features as shown in Tables 5 and 6.

VII. CONCLUSION

In this paper, we have demonstrated that Kim et al.'s and Hwang et al.'s DBAKA protocols for wireless roaming service are vulnerable to known key attacks. Furthermore, these DBAKA protocols for wireless roaming service fail to provide the communication confidentiality. Then we proposed a new DBAKA protocol for wireless roaming service based on ID-based cryptography. The proposed scheme eliminates all the vulnerabilities of the protocols [8], [10]. Moreover, we have conducted a BAN-logic analysis, which confirmed that our protocol satisfies mutual authentication. Detailed security analysis shows that the proposed scheme provides better security and more admired functionality features as compared with other existing DBAKA protocols for wireless roaming service. The proposed authentication scheme for TMISs requires a little more communication and computational costs than Tsai et al.'s protocol, but these computations are still lightweight, which is completely acceptable and applicable for the mobile device. Future work includes implementing the proposed scheme in a real-world environment and designing DBAKA protocols for wireless roaming service with strong untraceability.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their comments to improve the paper. They would also like to thank Prof. Jin Li from Guangzhou University for helpful discussions.

REFERENCES

[1] M. Rahnema, "Overview of the GSM system and protocol architecture," *IEEE Commun. Mag.*, vol. 31, no. 4, pp. 92–100, Apr. 1993.

[2] Y. Lu, L. Li, H. Peng, and Y. Yang, "Robust anonymous two-factor authenticated key exchange scheme for mobile client-server environment," *Secur. Commun. Netw.*, vol. 9, no. 11, pp. 1331–1339, 2016.

[3] M. Z. A. Bhuiyan, G. Wang, J. Wu, J. Cao, X. Liu, and T. Wang, "Dependable structural health monitoring using wireless sensor networks," *IEEE Trans. Depend. Sec. Comput.*, vol. 14, no. 4, pp. 363–376, Jul./Aug. 2017.

[4] W.-B. Lee and C.-K. Yeh, "A new delegation-based authentication protocol for use in portable communication systems," *IEEE Trans. Wireless Commun.*, vol. 4, no. 1, pp. 57–64, Jan. 2005.

[5] Y. Lu, G. Xu, L. Li, and Y. Yang, "Anonymous three-factor authenticated key agreement for wireless sensor networks," *Wireless Netw.*, vol. 11, pp. 1–15, Nov. 2017.

[6] Z. Cai, H. Yan, P. Li, Z. Huang, and C.-Z. Gao, "Towards secure and flexible EHR sharing in mobile health cloud under static assumptions," *Cluster Comput.*, vol. 20, no. 3, pp. 2415–2422, 2017.

[7] C. Tang and D. O. Wu, "An efficient mobile authentication scheme for wireless networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 4, pp. 1408–1416, Apr. 2008.

[8] M. Kim, N. Park, and D. Won, "Security analysis of a delegation-based authentication protocol for wireless roaming service," in *Multimedia and Ubiquitous Engineering* (Lecture Notes in Electrical Engineering), vol. 308. Heidelberg, Germany: Springer, 2014, pp. 445–450.

[9] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*, vol. 84. Heidelberg, Germany: Springer, 1985, pp. 47–53.

[10] S.-J. Hwang and C.-H. You, "A delegation-based unlinkable authentication protocol for portable communication systems with non-repudiation," in *Advanced Technologies, Embedded and Multimedia for Human-Centric Computing* (Lecture Notes in Electrical Engineering), vol. 260. Heidelberg, Germany: Springer, 2014, pp. 923–932.

[11] J. L. Tsai, N. W. Lo, and T. C. Wu, "Secure delegation-based authentication protocol for wireless roaming service," *IEEE Commun. Lett.*, vol. 16, no. 7, pp. 1100–1102, Jul. 2012.

[12] M. J. Beller, L.-F. Chang, and Y. Yacobi, "Privacy and authentication on a portable communications system," *IEEE J. Sel. Areas Commun.*, vol. 11, no. 6, pp. 821–829, Aug. 1993.

[13] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1953–1966, Sep. 2015.

[14] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, and Y. Tang, "Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks," *J. Netw. Comput. Appl.*, vol. 106, pp. 117–123, Mar. 2018, doi: 10.1016/j.jnca.2018.01.003.

[15] L. Pang, H. Li, X. Zhou, and Y. Wang, "A novel authentication scheme with anonymity for wireless communications," *Appl. Math. Inf. Sci.*, vol. 8, no. 6, pp. 3021–3026, 2014.

[16] J. Li, Y. Zhang, X. Chen, and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," *Comput. Secur.*, vol. 72, pp. 1–12, Jan. 2018.

[17] Z. Huang, S. Liu, X. Mao, K. Chen, and J. Li, "Insight of the protection for data security under selective opening attacks," *Inf. Sci.*, vols. 412–413, pp. 223–241, Oct. 2017.

[18] V. Odelu, A. K. Das, S. Kumari, X. Huang, and M. Wazid, "Provably secure authenticated key agreement scheme for distributed mobile cloud computing services," *Future Gener. Comput. Syst.*, vol. 68, pp. 74–88, Mar. 2017.

[19] J. Shen, T. Zhou, X. Chen, J. Li, and W. Susilo, "Anonymous and traceable group data sharing in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 4, pp. 912–925, Apr. 2018.

[20] G. Yang, Q. Huang, D. S. Wong, and X. Deng, "Universal authentication protocols for anonymous wireless communications," *IEEE Trans. Wireless Commun.*, vol. 9, no. 1, pp. 168–174, Jan. 2010.

[21] D. He, J. Bu, S. Chan, C. Chen, and M. Yin, "Privacy-preserving universal authentication protocol for wireless communications," *IEEE Trans. Wireless Commun.*, vol. 10, no. 2, pp. 431–436, Feb. 2011.

[22] H. J. Jo, J. H. Park, and D. H. Lee, "Efficient privacy-preserving authentication in wireless mobile networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 7, pp. 1469–1481, Jul. 2014.

[23] D. Zhao, H. Peng, L. Li, and Y. A. Yang, "A secure and effective anonymous authentication scheme for roaming service in global mobility networks," *Wireless Personal Commun.*, vol. 8, no. 1, pp. 247–269, 2014.

[24] H. Mun, K. Han, Y. S. Lee, C. Y. Yeun, and H. H. Choi, "Enhanced secure anonymous authentication scheme for roaming service in global mobility networks," *Math. Comput. Model.*, vol. 55, no. 1, pp. 214–222, 2012.

[25] X. Zhang, Y.-A. Tan, C. Liang, Y. Li, and J. Li, "A covert channel over VoLTE via adjusting silence periods," *IEEE Access*, vol. 6, pp. 9292–9302, 2018. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2018.2802783

[26] A. G. Reddy, A. K. Das, E.-J. Yoon, and K.-Y. Yoo, "A secure anonymous authentication protocol for mobile services on elliptic curve cryptography," *IEEE Access*, vol. 4, pp. 4394–4407, 2016.

[27] M. M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures: Delegation of the power to sign messages," *IEICE Trans. Fund. Electron.*, vol. E79-A(9), pp. 1338–1353, Jan. 1996.

[28] T. F. Lee, S. H. Chang, T. Hwang, and S. K. Chong, "Enhanced delegation-based authentication protocol for PCSs," *IEEE Trans. Wireless Commun.*, vol. 8, no. 5, pp. 2166–2171, May 2009.

[29] T.-Y. Youn and J. Lim, "Improved delegation-based authentication protocol for secure roaming service with unlinkability," *IEEE Commun. Lett.*, vol. 14, no. 9, pp. 791–793, Sep. 2010.

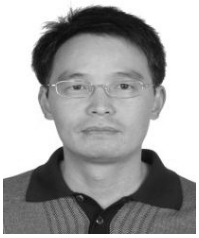
[30] C. C. Lee, R. X. Chang, T. Y. Chen, and L. A. Chen, "An improved delegation-based authentication protocol for PCSs," *Inf. Technol. Control*, vol. 41, no. 3, pp. 258–267, 2012.

[31] P. Gope and T. Hwang, "Security weaknesses on a delegation-based authentication protocol for PCSs," *Inf. Technol. Control*, vol. 44, no. 3, pp. 329–333, 2015.

[32] J.-Z. Lu and J. Zhou, "On the security of an efficient mobile authentication scheme for wireless networks," in *Proc. Int. Conf. Wireless Commun. Netw. Mobile Comput.*, Chengdu, China, 2010, pp. 1–3.

[33] J. Z. Lu and J. P. Zhou, "Preventing delegation-based mobile authentications from man-in-the-middle attacks," *Comput. Standards Interfaces*, vol. 34, no. 3, pp. 314–326, 2012.

[34] C. H. Wang and C. Y. Lin, "An efficient delegation-based roaming payment protocol against denial of service attacks," in *Proc. Int. Conf. Electron., Commun. Control*, 2011, pp. 4136–4140.
[35] Y. Wang, Q. Pu, and S. Wu, "Cryptanalysis and enhancements of delegation-based authentication protocol for secure roaming service," *Int. J. Electron. Sec. Digital Forensics*, vol. 4, no. 4, pp. 252–260, 2012.
[36] M. Abdalla, P.-A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Proc. PKC*, 2005, pp. 65–84.
[37] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, 1990.
[38] P. Syverson and I. Cervesato, "The logic of authentication protocols," in *Foundations of Security Analysis and Design* (Lecture Notes in Computer Science), vol. 2171. Heidelberg, Germany: Springer, 2001, pp. 63–136.
[39] C. P. Schnorr, "Efficient signature generation by smart cards," *J. Cryptol.*, vol. 4, pp. 161–174, Jan. 1991.
[40] J. Xu, L. Wei, Y. Zhang, A. Wang, F. Zhou, and C.-Z. Gao, "Dynamic fully homomorphic encryption-based merkle tree for lightweight streaming authenticated data structures," *J. Netw. Comput. Appl.*, vol. 107, pp. 113–124, Apr. 2018, doi: 10.1016/j.jnca.2018.01.014.
[41] H. Wang, Z. Zheng, L. Wu, and P. Li, "New directly revocable attribute-based encryption scheme and its application in cloud storage environment," *Cluster Comput.*, vol. 20, no. 3, pp. 2385–2392, 2017.
[42] Z. Tan, "An efficient identity-based tripartite authenticated key agreement protocol," *Electron. Commerce Res.*, vol. 12, no. 4, pp. 505–518, 2012.
[43] *PBC (Pairing-Based Cryptography) Library*. Accessed: Mar. 28, 2017. [Online]. Available: http://crypto.stanford.edu/pbc/
[44] *GMP (GNU Multiple Precision) Arithmetic Library*. Accessed: Apr. 18, 2017. [Online]. Available: http://gmplib.org/



ZUOWEN TAN was born in Yiyang, Hunan, China, in 1967. He received the M.S. degree in fundamental mathematics and the Ph.D. degree in applied mathematics from the Institute of Systems Science, Academy of Mathematics and System Science, CAS, in 2002 and 2005, respectively.

He is currently a Full Professor with the Department of Computer Science and Technology, School of Information Technology, Jiangxi University of Finance and Economics. He has authored over 70 articles. His research interests include big data security, information security, and cryptography.

Dr. Tan is a committee member of some international conferences and a reviewer of international journals.